# Adaptive Learning in an Educational Game

## Adapting Game Complexity to Gameplay Increases Efficiency of Learning

A master thesis for the title of *Master of Science* (*MSc*) in *Cognitive Artificial Intelligence* by


Jeroen Linssen

Utrecht University, Faculty of Humanities, Department of Philosophy


Dated 2011–08–11 for 30 ECTS

**Supervisors**

prof. dr. John-Jules Meyer from Intelligent Systems
dr. Herre van Oostendorp from Human Media Interaction
Erik van der Spek, MSc from Human Media Interaction

Game research for training and entertainment

Universiteit Utrecht


**Keywords:** adaptive learning, adaptivity, artificial intelligence, educational games, game design, serious games

## Abstract

This thesis investigates the possibilities of adaptivity in an educational game called *Code Red: Triage*. This game lets players assume the role of a medical first responder who has to triage victims. A triage is a procedure through which medical personnel can determine the priority of a victim.
The first hypothesis of this thesis is that, by making the game adapt itself autonomously to the player, it becomes more efficient. That is, based on how a player performs in the game, the game changes its properties to suit his needs. The second hypothesis is that the player feels more engaged by the game if it adapts itself to his needs. This would be the case because he would be challenged optimally by the adaptations the game makes.
The adaptation consists of determining which victim cases should be presented to the player. Each victim has his own complexity, so when a player reaches a particular skill level, slightly more complex victims are presented to him.
This idea is implemented in Code Red: Triage and subjected to an experiment in order to falsify the hypotheses. The experiment is comparitive in nature and features two groups: one playing the adaptive version of the game and one (the control group) playing the non-adaptive version.
Results from the experiment confirm the hypothesis concerning efficiency: adaptivity in the game lets players gain knowledge faster, i.e., in less time and in less victim cases. Thus, adaptive learning is more efficient than static learning. However, engagement was found not to differ between conditions, disproving the second hypothesis.
Nonetheless, these outcomes indicate that adaptivity has promising possibilities for educational games by saving time and effort of the players. Future research should investigate in different possibilities of in-game adaptivity that heighten efficiency or engagement.

# Preface

About a decade ago, I was seeking forms of entertainment for my personal leisure and discovered a domain that was both exciting and diverse. Indeed, it was the domain of *computer games*. What I found compelling about such games was that actions you take often provoke direct reactions. That is, when you reach a certain goal, you directly receive a reward – a nice abstraction from real life, especially if the reward is as something transparent, like a score. Logically, such behaviour stimulated me (and other players) to continue playing the game to earn more rewards.

Over time, I began to appreciate computer games more and more. As my understanding of their characteristics evolved, so did their characteristics evolve themselves. Increasingly more effort was put into the creation of games as more and more design choices were made to present them in their final forms. One of these choices that has been around since the beginning of the computer game era is the implementation of the *difficulty* of the game. This is important because as players progress through a game, their skills in playing it will increase and have to be matched to keep the game interesting enough. After all, if a game does not offer any challenge, it becomes superfluous to earn the rewards the game offers for completing tasks. I highlight this notion because it is strongly tied to this thesis, which becomes apparent during the following paragraphs.

While games are usually thought to be no more than 'time killers', I was interested to investigate the possibilities games can offer for more serious purposes, i.e., *serious games*. This branch of gaming focuses on training the player by practising a procedure or by helping him to attain knowledge. The study matter is incorporated in the game and the player has to perform tasks based on that matter, for which he receives feedback and rewards. By doing so, he *learns* while playing the game. The idea behind games-based learning is that players have a higher incentive to continue learning – for instance, their progress can be visualized in the form of a score to make it more tangible.

For my master's thesis, I went in search of projects connected to the Utrecht University that dealt with serious games. The first option that came to mind was one that I remembered from earlier on and I was reminded about while carrying out my internship at TNO Soesterberg. The project I remembered is called the *GATE project*, which is funded by the Dutch government and seeks to research games and game-technology while also thinking of ways to convert and implement gained results in practical applications. Browsing their website, I found out about the existence of different work packages, of which I mailed the supervisors to inform me about the details of their respective work. In the end, the most interesting option for me was offered by Herre van Oostendorp and Erik van der Spek, who work on improving the design of serious games from a cognitive standpoint. This is how I became acquainted with Erik's game *Code Red: Triage*, which I had the honour to use for this thesis.

How the story continued from there becomes apparent in sections 1 to 7. I hope this turns out to be an interesting read; I certainly did my best to maintain a clear train of thought. If some sections are confusing at first, try to re-read them or turn to the glossary for an explanation of particular keywords.

# Acknowledgements

A word of thank I must utter to my supervisors Herre and Erik, who gave sound critique on my work at regular intervals during the past six months. Without these meetings and their support, I fear this thesis would be, among other things, less structured and less clear.

My thanks also go to all those who reacted enthusiastically when I explained my thesis subject and motivated me to make it worth my (and their) while.

Lastly, as a form of gratitude, but also as a guide for the reader and, in particular, prospective thesis writers, I present a list of software that I used for the creation of this document and for carrying out the experiment and interpreting the results. Most of these are freeware (whereof some proprietary) and some commercial – though the free 30-day license on SPSS 17 proved to be sufficient.

**Text:** LyX 2.0, LibreOffice Writer 3.4, Notepad++ 5.8.7

I never delved into LaTeX, because LyX struck me as a good WYSIWYM (sic) editor for TeX documents. It is not complete in its functionality, but it makes a lot of things more straightforward and more clear. The LibreOffice suite is a good alternative for Microsoft Office for processing small documents and Notepad++ is an excellent text editor, particularly because of its regular expression plug-in.

**Graphics:** LibreOffice Draw 3.4, Inkscape 0.48, Gimp 2.6

Most of the diagrams in this document were made with LibreOffice Draw, some were tinkered with in Inkscape and Gimp was used for resizing screenshots.

**Statistics:** SPSS 17, LibreOffice Calc 3.4

While LibreOffice Calc was easy to use for some baseline statistics functions, SPSS 17 offers much more of those in a much simpler way.

**Coding:** Microsoft Visual C++ Studio 2008 Express, Source SDK

Of course, I could not have implemented adaptivity in Code Red: Triage without performing some coding. Microsoft's Visual C++ Studio 2008 Express edition was free to use and offered all that I needed. Of course, Valve's Source SDK was used to carry out alterations in the game's levels.

# Glossary

| Term | Description |
|---|---|
| *Adaptive condition* | The condition in the experiment in which players play the adaptive version of *Code Red: Triage* |
| *Adaptivity* | The autonomous alteration of certain properties |
| *Breathing frequency* | Number of breaths a person takes per minute |
| *Capillary refill time (CRT)* | The rate at which blood refills empty capillaries; used to indicate dehydration |
| *Code Red: Triage* | COgnition-based DEsign Rules Enhancing Decision-making TRaining In A Game Environment; name of research project and educational game |
| *Complexity (of a victim)* | Smallest number of successive checks or actions that need to be performed to *triage* a *victim successfully* |
| *Control condition* | The condition in the experiment in which players play the non-adaptive version of *Code Red: Triage* |
| *Difficulty (of a victim)* | Subjective measure of effort it takes to *triage* a *victim*; *complexity* is used as an objective term |
| *Dynamic complexity adaptation (DCA)* | Term used to denote the autonomous adaptation of a game's *complexity*; used instead of *DDA* to emphasize objectiveness |
| *Dynamic difficulty adaptation (DDA)* | Term used to denote the autonomous adaptation of a game's *difficulty*; see *dynamic complexity adjustment* |
| *Educational (or serious) game* | Game designed to teach people about a subject |
| *Efficiency* | The amount of learning per time unit (e.g., per second or per *victim*) |
| *Flow* | Psychological concept: a combination of skill level and challenge through which a person can perform a task optimally |
| *Score* | Number that indicates how good the *triage* was performed, ranging from 0 to 100 |
| *Successful triage* | A *triage* performed by the player with a *score* that is at least as high as the *threshold* for the *tier* of that *victim* |

## Glossary (continued)

| Term | Description |
|---|---|
| *Threshold* | Minimum *score* a player has to reach on a *triage* to continue to the next *tier* |
| *Tier* | Level of *complexity* of a *victim* |
| *Triage (procedure)* | Procedure through which medical personnel determines the priority of *victims* |
| *Victim* | Person in need of care; entity in game world that the player needs to *triage* |

# Contents

# 1 Introduction

In 2007, Van Oostendorp and Van der Spek, in cooperation with Pieter Wouters, began a project to counter the lack of empirical scientific research in serious games. Their first step was the creation of a game called *Code Red: Triage* (COgnition-based DEsign Rules Enhancing Decision-making TRaining In A Game Environment, or *CRT*), which was set up to train players in performing *triages*. A triage is a procedure performed on an injured human being through which medical personnel can determine in how much need of help that person is. The game features a 3D environment in which the player controls a medical first responder whose job is to find victims and triage them. Each victim has his or her specific injuries and it is up to the player to determine (1) which actions and checks to perform on the victim and (2) to what category that victim belongs. Code Red: Triage gives feedback about each victim and awards the player with a score pertaining to the triage procedure.[1]

The main research question concerning Code Red: Triage is which factors influence the learning of the player. Previous experiments carried out by Van der Spek focused on the presentation of auditory and visual cues and the gradual increase of possible actions and checks that were available to the player while performing a triage. Regrettably, the implementations that were chosen did not yield much significant results in experiments. A different line of thought arose that appealed to the inherent nature of games, which is their high level of interactivity. More specifically, it can be said that games are able to *adapt* themselves to the player. This idea can be extrapolated to Code Red: Triage in the form of *adaptively* presenting victims a player triages to his skill level. That is, by implementing adaptivity in the game, the game itself autonomously decides which victim the player triages based on his prowess. In this manner, the difficulty of the game is tailored to the skill level of the player.

There seems to be ample research in the field of in-game adaptivity performed in the scientific world. Therefore, I took up the challenge of researching adaptivity in Code Red: Triage – as is apparent, the results of my research are detailed in this thesis. The focus of my research lies on the determining of the effects of adaptivity in the game on the *efficiency* of learning and *engagement* of the game. Because adaptivity changes game properties based on input from the player, certain tasks in the game that the player has already mastered can be omitted, resulting in more efficient gameplay. Additionally, omitting such trivial tasks may lead to a higher level of engagement, because the player has to face challenges that are suited to his skill.

My thesis consists of researching the best possible ways of incorporating adaptivity, implementing them and empirically testing my hypotheses in a two-group experiment – one group playing the adaptive, the other group playing the non-adaptive version of the game. The research question and hypotheses I adopted follow straightforwardly and are written down below.

**Research question:** Does adaptivity make learning in Code Red: Triage more efficient?

**Hypothesis 1:** In the adaptive version of Code Red: Triage, the player is able to learn more efficiently. This is the case because the game decides when the player is able to move on the next difficulty, possibly shortening the amount of time taken to finish the game, but letting the player retain the same amount of knowledge.

**Hypothesis 2:** The player feels more engaged by the adaptive version of Code Red: Triage, because the game always keeps being challenging.

This thesis follows the following format. The next section focuses on the game itself, explaining how it works. The third section details literature research that was performed, while section four and five treat the design and implementation of adaptivity in the game. Section six treats the experiment that was carried out and finally section seven wraps it all up by providing a general conclusion, some points of discussion and an incentive for future research.

[1] Similar research has been conducted by Jarvis & De Freitas, [32], which was more focused on how feedback should be given to the player and much less about designing guide lines.

# 2 Code Red: Triage, A Serious Game

In 2007, Erik van der Spek, Pieter Wouters and Herre van Oostendorp set up a project to construe a set of design guidelines for serious games. They felt that the training of medical officers could be performed in such games because a game-environment can be made highly interactive and realistic, thus closely mimicking the real life experience. There has been some earlier research on serious games, but Van der Spek et al. noted that there was still a lack of empirical research, [45]. To perform new empirical research, a serious game called *COgnition-based DEsign Rules Enhancing Decision-making Training In A Game Environment* – or, acronymed, Code Red: Triage – was developed by Van der Spek as a testbed.

Several experiments have been carried out by Van der Spek pertaining to the addition of auditory and visual cues to the game and the variation of the number of actions and checks that are available to the player while triaging, see [42, 43, 44, 45]. This thesis strives to be a further exploration of techniques that can be used in serious game design.

The remainder of this section is split into two parts, the first dealing with the actual triage procedure as performed by medical personnel; the second explaining the details of the game.

## 2.1 Triage

As explained in the introduction, the game revolves around training the player to perform so-called triages. A triage is a procedure through which medical first responders can determine the priority of victims in case of an emergency. They can do so by following a series of tests – for instance, if a victim can still walk, he has a low priority, because his wounds are not very severe. In another occasion, where a victim can not walk, can still breathe, but has a breathing frequency which is too high, he is triaged as having a high priority. More specifically, there are four triage classes, viz. *T1*, *T2*, *T3* and *Dead*, in descending order of priority. Victims who are still mobile always fall in class T3, those who are not mobile, but whose other bodily functions are normal, fall in T2 and victims who have more severe injuries fall in class T1. When a victim seems to be dead, he has the lowest priority – resuscitation is not the task of the medical first responder.

In Figure 1, a flowchart showcasing the triage procedure is shown, based on [28].

The four classes are visible, as well as five checks and three actions that can be performed. *Mobility* is the property of a victim that is always checked first; as stated above, if the victim passes this check, it falls into category T3. If not, the victim's *airway* is inspected for a possible blockade. When this is the case, the medical first responder chooses one of two options: (1) performing a *chin lift* or (2) performing a *jaw thrust*, both of which have the effect of freeing the airway. The latter action is carried out instead of the former when the victim seems to have a neck trauma. If the victim does not start breathing after removing the blockade, he is considered to be (near) death and classified as such. When the medical first responder succeeds in letting the victim breathe again, he is put into a *stable side position* and classified as having a high priority, namely T1.

On the occasion that the victim's airway was not blocked, his *breathing frequency* needs to be checked. If this turns out to be too high or too low, i.e., above 29 or below 9 respirations per minute, the victim is classified as T1. If the frequency value lies between 9 and 29 respirations per minute, the next check to be brought about is whether his pulse is at a satisfactory level. This can be done in two ways as well – the fastest and easiest way to do so is to check the *capillary refill time*.[2] This amounts to applying pressure to the nail bed of one of the victim's fingers and determining whether the blood flows back fast enough, i.e. within 2 seconds. If this occurs, the victim does not have extreme injuries and is classified as T2. Otherwise, he is classified as T1. There is one drawback to this, namely that of the prerequisite of a good light source, because without one, the nail bed can not be observed. The alternative is a slower method: taking the *pulse* of the victim. If this is found to be below 120 bpm, the victim is classified as T2, otherwise, he is classified as T1.

[2] Acronymed as CRT; not too be confused with the meta-acronym of Code Red: Triage.

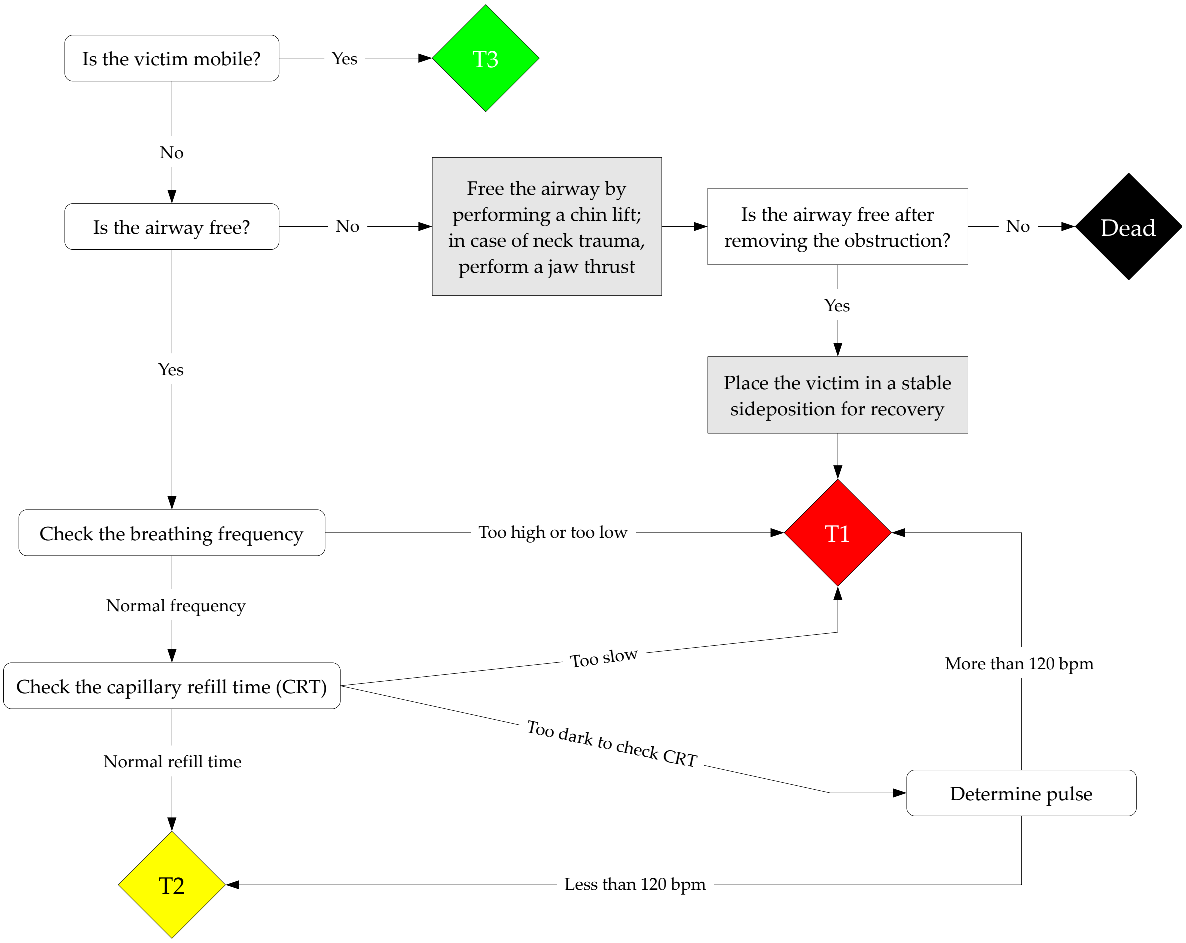



**Figure 1:** *Flowchart for the triage procedure. The diamonds denote the different categories, the grey rectangles* actions *and the rounded rectangles* checks.

## 2.2 The Game

In Code Red: Triage, the player assumes the role of a medical first responder charged with investigating a metro station that has fallen victim to a terrorist strike. A development kit called the Source SDK was used by Van der Spek for the creation and modification of the game's levels. This kit was developed by Valve Corporation and used for a variety of games – Half-Life 2, [57], being the prime example.

The game commences with some instructions indicating that there was a terrorist strike on a metro platform and that the player has to reach that platform and triage the victims present there. He enters the main lobby, see Figure 2, and has to find his way around the complex to the metro platform. The entrance to the metro is indicated and when the player travels down the stairs a new level loads, which consists of a series of hallways, including some objects which obstruct the passage, see Figure 3. The time it took the player to reach the metro platform through this level can be seen as an objective measure of the player's affinity with movement in 3D games.

**Figure 2:** *The main lobby of the game.*

**Figure 3:** *The hallway level.*

**Figure 4:** *The metro platform. Several victims can be seen lying on the ground.*


**Figure 5:** *Triaging a victim. Left and right of the victim are the check and action buttons, in the bottom the classification buttons and in the centre the information screen.*


**Figure 6:** *Information is presented after selecting the Jaw Thrust action.*


**Figure 7:** *Feedback on the performed triage.*

Upon having walked through this second level, a new level loads which contains the metro platform and all victims, see Figure 4. A timer counts down from 17 minutes to indicate how much time the player has left to complete his mission of finding and triaging all victims. To perform a triage, he is required to walk up to a victim and press the *E*-key to call up an overlay screen, see Figure 5, that includes eight buttons for all checks and actions the player can carry out and four buttons for the classifications. The centre of the screen is reserved for information about the victim – at first, when the player selects to triage a victim, general information is shown. When the player selects a check or an action, information about that step and its effect is shown, see Figure 6. When the player is confident that he has gained enough information, he can choose to classify the victim as one of the four classes: T1, T2, T3 or Dead.

Having classified the victim, the game determines whether the correct classification was chosen and gives a score to the player for this particular triage. This score ranges from 0 (wrong answer) to 100 (perfect procedure, including categorization). The latter can only be reached by choosing the correct classification and performing the correct checks or actions in the right order, within a preset time limit for the victim. Points are deduced from the perfect score for any deviation from the correct procedure and for taking more time to complete the triage than the time limit. The victim score is added to the total score, which is displayed at the top of the screen and visible in all screenshots featured here.

Feedback is given to the player in the form of a small text message in the game, indicating the score he reached on the victim as well as any missteps he made in the procedure. See Figure 7 for an example.

When the player has found all 19 victims or the timer has reached its end after 17 minutes, the game ends with a *GAME OVER* message detailing the number of victims he found and the total score he attained.

# 3 Adaptive Learning

This section treats the concept of *adaptive learning* by discussing both its theoretical aspects as well as actual implementations in order to gain knowledge about techniques that could be beneficial for implementing adaptivity in Code Red: Triage.

## 3.1 Adaptivity

By itself, *adaptivity* is a non-specific term: it simply denotes the possibility of change in accordance with other factors. It is an automated process in which something is able to alter itself in order to 'fit' into its surroundings. On a macroscopic scale, evolution could be thought of as an example of adaptivity, because it is the act of organisms adapting to their surroundings. The diminishing of an eye's pupil on receiving a higher amount of light is an example on a smaller scale.

The process of *learning* seems to be a much more complicated business than adaptivity, for it involves acquiring new or modifying existing knowledge, which can be performed through a variety of methods. Education in schools, universities and other course-based institutes helps people to learn, but the approaches used by these organizations are more often than not directed at a group of people. It follows from this fact that the way in which the material is offered to the students must give them all the possibility to understand the material. That is, the form of education has to be general enough in order to cover the needs of all individual students. However, each individual student can benefit from a learning approach that is tailored to his needs. For example, one student might have a better developed understanding of logic, while another student may a somewhat less developed understanding, which may influence their comprehension of the study material. Ultimately, the goal is having completely personalized learning methods for each student so that their learning efficiency may be optimized. Exactly this is the subject of the field of adaptive learning: developing an on-demand learning technique in which the learning environment adapts itself to its user's needs. This field of study began in the 1970s, detailed in a book about adaptive learning for children, see [5, 25].

## 3.2 Flow

The Hungarian psychologist Csíkszentmihályi Mihály asserted that activities can be performed optimally when a person is in the *flow*, [19], this being a mental state in which the subject is completely focused and motivated. A person can enter this immersion in his activity – in this case, learning – when his skill level and the challenge level of the material are at balance. That is, he can attain it when his skill level is high while the challenge level is high as well, as portrayed in Figure 8.

Pertaining to flow in education, Csíkszentmihályi noted that the Montessori Method (a method of nursery education in which children are allowed to develop themselves at their own pace) is able to provide better circumstances for attaining the flow state than traditional educational settings, based on results from [48]. Furthermore, an important thing to note is that one of the conditions to be met in order to achieve the flow state is the availability of direct feedback, [20]. In other words, a system that provides feedback to its user can help him reach the flow state, thus boosting the effectiveness of his activity.

Games can be said to be an implementation of such a system, for they interact with the player by giving feedback on his input, e.g., presenting a score when an in-game task is completed. Jenova Chen considered Csíkszentmihályi's idea for a game aptly called *flOw*[3] and construed Figure 9 to show how adaptations may keep the player in the flow over time. Being in the flow implies that the person in question is continuously challenged, which entails that he also has a higher level of engagement. Other implementations that are based on flow are discussed in section 3.5.

For this thesis, a form of adaptivity is implemented in Code Red: Triage in order to let players attain and sustain the flow level to improve the efficiency of their learning. For this cause, the next

[3] Discussed in section 3.5.5.

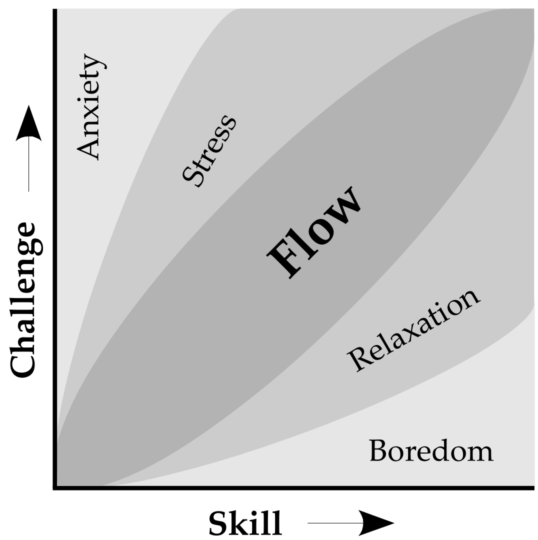



**Figure 8:** *Mental states as combinations of challenge and skill level.*

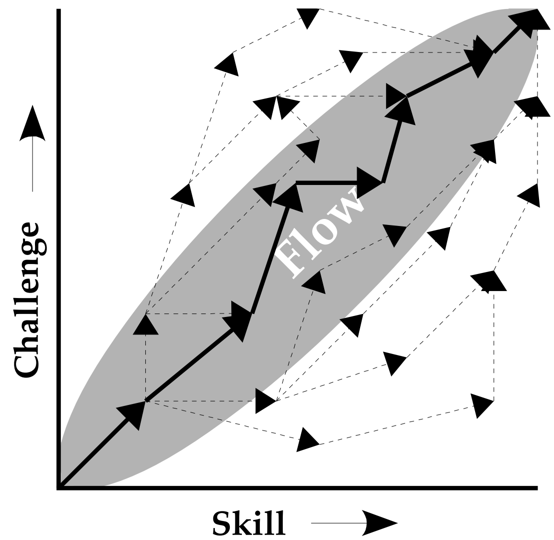



**Figure 9:** *Adaptation guides a player (bold arrows) in order to keep them 'in the flow', adapted from [15, p. 32]. Dashed arrows indicate other possible actions.*

sections provide an overview of the concept of adaptivity in literature.

## 3.3 Adaptive Education

The idea of flow has already been used by several researchers to create systems that assist their users in an adaptive way. One such system is called the ALIGN (Adaptive Learning In Games through Non-invasion) system, developed and implemented in an educational game by Peirce et al., [46]. They state that the concept of flow is something game design and educational adaptation have in common, because of the balancing of challenge and skills. The general setup of ALIGN was intended as a framework for adaptation in (serious) games. While being basic, it describes the use of a feedback loop that is divided into several distinct modules. This loop consists of (1) processing raw data from the game; (2) constructing a model of the user, incorporating information about the state of the game and previous adaptations; (3) creating possible adaptations and (4) applying selected adaptations.

A more abstract form of such user-adaptive system (UAS) schemas can be taken from Anthony Jameson's work, [31], as can be seen in Figure 10. In this figure, the slanted U stands for the user, while ovals denote input or output; rectangles stand for processing methods and the cylinder stands for stored information. The dashed arrows designate use of information and the solid arrows indicate the production of results. The dotted arrow is an addition on the original schema that denotes that it is repeated loop-wise.

The ALIGN framework is an extension of this schema, instantiating the various processes. That is, in ALIGN, the information about *U* is the raw data taken from the game and the decisions about *U* stand for the adaptations that are made. Furthermore, the schema is enclosed in a loop, because the adaptation is repeated indefinitely (the dotted arrow).

An implementation of the ALIGN mechanic was evaluated by Peirce et al. in the ELEKTRA (Enhanced Learning Experience and Knowledge TRAnsfer) project, [1], which was an interdisciplinary venture into technology-enhanced learning. In the experiment, players had to overcome challenges in the form of physics puzzles that were actually part of their assessable curriculum and did so in two subsequent attempts. The first attempt was, for every participant, the ELEKTRA game without

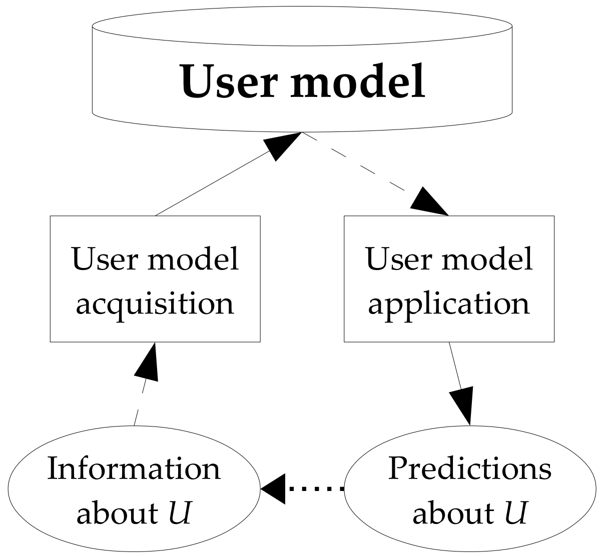



**Figure 10:** *General schema for processing in a user-adaptive system, adapted from [31, pp. 306], with addition of dotted arrow.*

hints, while the second attempt combined ELEKTRA with either neutral (out-of-context), adaptive, counter-adaptive (for instance, a hint about the density of an iron marble when dealing with a plastic one) and no hints. The measure for the experiment was the difference in absolute distance to the correct solution (on a 100-point scale) based on the two attempts. Means that were found for neutral, counter-adaptive and no hints were approximately 4, whereas the mean found for adaptive hints was almost 5. Peirce et al. acknowledge that this positive outcome was not statistically significant, yet still encouraging for further research.

Another adaptive learning system is *AutoTutor*, developed at the Institute for Intelligent Systems at the University of Memphis. Having started in 1994, a collection of researchers under the guidance of Arthur Graesser have built an intelligent tutoring system that is able to converse in natural speech with a student, see [26]. On top of this system, an animated agent which has facial expressions and ability to gesture was designed. It assists students in learning about Newtonian physics, computer literacy and scientific reasoning by responding to the questions the students ask – it is said to take 50 to 100 turns between the student and the agent for the collaboration to converge on a good answer. In this manner, the student can be seen as having a personal tutor that provides the correct personal learning method. Results from various experiments indicate an increase in learning with a mean of 0.8 point on a 1 to 10 scale, see [27].

These tutoring systems, [3, 60], rely for a great part on self-explanation by responding to the student with questions about his reasons for choosing an answer, thus leading to deeper understanding of the material. AutoTutor implements this behaviour as well, but it also adapts its learning plan on-the-fly to the answers the student gives and, correspondingly, to the skill level of the student. It is able to do so through use of the *frontier learning* principle, which consists of focusing on the conversation on material which is not yet understood as good as desired, and the *coherence* principle, responsible for selecting the subject that is most similar to the previous subject. AutoTutor uses Latent Semantic Analysis for evaluation of answers, see [35]. A version of AutoTutor that is being developed, see [22], also incorporates adaptation to learner emotions, for instance by engaging the student more when he seems bored or by giving more information when he seems puzzled.

In summary, Graesser et al.'s tutoring system reacts to a student's inferred state – his level of understanding or his (emotional) state – by choosing an action, i.e., asking a question, that corresponds to the student's state in order to improve his understanding.

**Benefits for the implementation** Jameson's schema for user-adaptive systems serves the abstract design of the implementation of adaptation in Code Red: Triage, with Peirce et al.'s ALIGN framework showcasing an instantiation of the schema. As the showcased systems rely on human-computer interaction through dialogue, their workings are not directly applicable to Code Red: Triage.

## 3.4 Adaptation in Commercial Games

All commercial computer games (*games* from here on) are prone to long periods of play-testing in order to determine whether they do not have bugs, exploits or imbalanced play. The latter is of extreme importance, because if a game is too easy, it will have little replay value and little challenge – if it is too difficult, players may quit playing before reaching the end of the game in frustration. Almost all contemporary games incorporate some option to scale the difficulty for players, yet, as Spronck, [54], also noted, actively thinking about a meta-game concept like the difficulty level obstructs immersion. Some game developers noted this as well, for instance Scott Miller, owner/partner of the game company 3D Realms, who strongly advocated the use of *dynamic difficulty adaptation*[4] (DDA) in the 2001 third person shooter game *Max Payne*, [49]. The discussion in [39] concerning Miller's points and the implementation of DDA in Max Payne shows that not all players were very content with it, because of the possibilities to exploit the game as well the incompleteness of the algorithm used. For instance, players could repeatedly let themselves die, which would result in an overall decrease of difficulty. Several players noted that they often saved[5] after and before difficult fights and would reload immediately if their health dropped – in this manner, they could accomplish defeating tough enemies without the game noticing them dying. They felt that this undermined the idea of DDA because enemies became harder to defeat and were of the opinion that the algorithm should have included taking into account of the number of saves and loads of game. However, these two criticisms can be labelled as being faults on the player's behalf, because it is their erroneous meta-game thinking that constitutes them.

A more concrete critique is that players can feel cheated when the changes made by DDA are obvious to them. This was the case in several games, for instance in *The Elder Scrolls III* & *IV* and *FallOut 3*, [8, 9, 10], the player roams a giant world and can increase his statistics, e.g., his strength, weaponry, but the difficulty of the enemies he faced was scaled to his level. This results in costing the player as much effort as it cost him at the beginning of the game to kill some rats as it would in the penultimate part of the game, see [2, 47, 38]. This emphasizes the need of not adapting all the elements of a game. There are also games that knowingly make the adaptation transparent, e.g., the beat 'em up game *God Hand*, [18], which implements a visible bar that shows the difficulty of the enemies the player faces. This bar also shows the progress the player makes towards the next or previous difficulty level, see [52].

Other games have included crude examples of DDA as well, for instance *Guitar Hero 5*, [40], an instrument simulation game, in which the player needs to hit the correct notes of a song with good timing. The difficulty adaptation used in this example relied on the number of notes played correctly, adding more notes and paying more attention to timing when the player did well and inverse. Racing games like *Mario Kart* and *Need for Speed*, [41, 23], implemented a simple adaptation which is known as 'rubber banding': when the player lagged behind the other racing contestants, they would slow down in order to let the player catch up with them – when the player was up front, his opponents would become faster and tried to keep up with him. As in Max Payne, the first person shooter *Sin Episodes – Emergence*, [50], varies the skill, numbers and toughness of enemies based on the state of the player or the change thereof.

One of the most complex implementations of dynamic difficulty adaptation in games is found in

[4] In this thesis, the term 'dynamic difficulty adaptation' is used, as opposed to other literature on games, where 'dynamic difficulty adjustment' is used; the latter could arouse confusion, as it is not clear by who the adjustment is made, whereas 'adaptation' implies an automated process. Other notations for this concept are 'difficulty scaling' and 'auto-dynamic difficulty'.

[5] *Saving* in games stands for saving the game world at a particular point, including all properties like player health, location, items and enemies. Complementary, *loading* stands for loading a save and thus returning to that game state.

the *Left 4 Dead*, [58, 59], series: first person shooters in which a team of up to four persons have to cooperate to overcome hordes of zombies. In these games, a module called the *AI Director* controls the placing of enemies in varying positions and numbers based on the status of the players, while also controlling placement of beneficial items like health packs, weapons and ammo. Yet the players are offered a choice beforehand, as they are still able to select a difficulty level – their choice influences the adaptations made by the AI Director, making it spawn fewer items and stronger enemies more frequently if a higher difficulty is chosen. Thus, players can make a base choice in difficulty which still leaves room for further deviation from that difficulty.

The next section discusses ideas on dynamic difficulty adaptation in games in scientific literature, which provides better grounds and constructions for approaches.

**Benefits for the implementation** The discussed examples showcase adaptivity applied to a variety of games, but the implementations are often very crude or not entirely applicable to the domain of serious games, especially because of the lack of a player-versus-enemy setting therein.

## 3.5 Formal Approaches

This section discusses the scientific research that has been performed into the idea of adapting a game's difficulty to players' abilities.

### 3.5.1 Dynamic Difficulty Adaptation

Christine Bailey presents her ideas on *dynamic difficulty adaptation*[6] – changes made autonomously by the game concerning the difficulty – in a general fashion, starting with a summary of the elements of a game that can be adapted, [6, p. 2], which is repeated below.

**Player character attributes:** Specific properties of the player, e.g., speed of movement, health and attack damage.

**Non-player character attributes:** Specific properties of all non-player characters (NPCs), namely all player character attributes and also the decision-making processes and aiming and path-finding qualities.

**Game world and level attributes:** The structure and content of the game world can be adjusted, for instance by placing health packs and ammunition or by making jumping over gaps easier by making the gap smaller.

**Puzzle and obstacle attributes:** Incorporating a dynamic element in mostly static puzzles may sometimes not be possible, but the results from earlier solved puzzles can influence the adaptation of later ones.

This abstraction can easily be used to see which elements are candidates for adaptation. Bailey asserts that two questions must be answered before an adaptation algorithm can be implemented, namely *'When?'* and *'How?'* These are not difficult questions, for the answer to the former is a measure of the current skill level of the player, while the latter is based on the type and emotional state of the player, which is reminiscent of the difficulty levels used in the Left 4 Dead series (see the previous section).

A testbed for Bailey's ideas has been developed in the Unreal Engine 2.0, from Unreal II: The Awakening, [36], but, regrettably, was only evaluated in an informal manner. This testing phase showed that while the mini-games (simple maze navigation and dodge games) were suitable for DDA, but were lacking in scope, because test subjects found them to be very repetitive. Bailey noted that this could be overcome in the Neomancer project, [33], a role-playing game similar to the Elder

[6] In her words, "auto-dynamic difficulty", but in this thesis, a general term is used to do away with the different terms used in the literature.

Scrolls series, in which the player character must traverse a continent and solve quests. This project was led by Michael Katchabaw, but has not been developed any further as well.

A more concrete approach to the concept of dynamic difficulty adaptation[7] was construed by Robin Hunicke and Vernell Chapman, who created a DDA system called *Hamlet*, [29, 30], implemented using the *GoldSrc* game engine from *Half-Life*, [56]. Hamlet's functionality consists of monitoring game statistics, defining and executing adjustment actions and policies and generating play session histories. Hunicke and Chapman describe the goal of Hamlet as trying to prevent the player reaching a state in which he can not successfully complete the game, [30, p. 92]. They also adhere to the principle of flow – keeping the player in a state that is optimal for his experience, i.e., presenting an apt challenge.

A repeated lack of resources (ammunition for his gun, health packs) a player has to have in his possession to continue in the game can indicate the inability of that player to complete the game. Thus, Hamlet tries to prevent this by choosing a reactive adjustment, like manipulating the strength of weapons or the health of enemies, in the scene the player currently inhabits or by performing a proactive adjustment, e.g., changing properties of item or enemies that the player does not see. Furthermore, Hamlet can create difficulty levels which Hunicke and Chapman call zones, that try to keep a player at a specific health level, e.g., 25% or 75%. Bailey mentioned this as well, indicating the need of a sort of meta-adaptation: the adjustment of the adaptation algorithm to the player. In the former case, enemies increase in difficulty quite fast and ammunition and health packs are dropped rarely, while the latter incorporates the inverse.

For an experiment, a 'comfort' zone policy for Hamlet was implemented, which tried to keep players' health at a value of 60% with a standard deviation of 15%. Participants played a game called *Case Closed*, [21], for 15 minutes, with one half of the group playing an unadjusted version, and the other half playing the Hamlet version. In the unadjusted version, players died an average of 6.4 times, while in the Hamlet version players only died an average of 4 times. However, taking the number of deaths of the player as a measure is a harsh indication of the effectiveness of the adaptation, for the lower number of deaths in the adapted version of Case Closed could just as well indicate a lower difficulty.

Trends also showed that expert players reported slightly higher levels of enjoyment, where novice players did not. This could be the case because expert players knew better what to expect of the game beforehand and could thus compare the effects of the adaptation, though subjects often reported adjustments that were not there or gave no notice of adjustments that really had happened.

**Benefits for the implementation** Bailey's summary of elements that can be adapted seems to be a convenient list, yet it is not completely straightforward. For instance, in Code Red: Triage, the difficulty of the victims that are to be triaged can be placed in the category of NPC attributes, but it can just as well be placed in the category of puzzle attributes. Thus, Bailey's idea of such a division of game elements is not convenient for games in which these elements are tightly mingled.

Implementing Bailey's thoughts about DDA, Hunicke and Chapman's Hamlet system is able to tailor the properties of the player character, the NPCs and the environment. However, the measure chosen by Hunicke and Chapman is a doubtful one, which undermines their conclusions. Still, this is an incentive for more proper research, which is performed in this thesis.

### 3.5.2 Dynamic Scripting

Discontent with the abilities of contemporary game AI, Pieter Spronck researched the possibilities of using machine-learning techniques to increase the quality of complex game AI. His 2005 dissertation, [54], includes the topics of off- and online learning and to what extent they can be used to increase the effectiveness of game AI. The difference between the two is that online learning happens during

[7] Hunicke and Chapman use the 'dangerous' term "dynamic difficulty adjustment", which leaves open exactly *who* adjusts the difficulty.

gameplay, while offline learning has been done in the testing phase of a game. The former is of importance to this thesis, because it is the generalized form of adaptivity in games.

A part of Spronck's research was devoted to developing a novel approach called *dynamic scripting*, which was designed to meet all these requirements. In computer games, scripting is a static technique used to trigger events – for instance, if a player reaches some part of a level, a cut-scene in which a part of the story of the game is detailed ensues. The key ability of dynamic scripting is that it manages rulebases that construct behaviour of the game characters on-the-fly and updates this behaviour based on previous experiences.

For example, an implementation of dynamic scripting called *MiniGate*, based on the extensive core mechanics of the games in the *Baldur's Gate*[8] series, [11], was evaluated by Spronck in order to test its efficiency. In this game, two teams with the same team members were pitted against each other in battle, where one team used static tactics that are deemed strong and the other team made use of dynamic scripting to adapt their tactics. Typical tactics for game characters could consist of, for instance, first drinking a strength potion that would increase their attack damage, after which they would attack. Dynamic scripting was used by one team to construct effective tactics, that is, over time, they learned which actions they should take in which order to defeat their opponents. In a series of experiments, the team using dynamic scripting was able to attain a higher fitness over time.

Because dynamic scripting makes automatic changes in small steps that affect the computer's tactics, Spronck asserts that it may overcome issues usually found in static games, such as coarse difficulty settings and the manual choice of difficulty. Furthermore, by limiting the updating functions used for new behaviour, this approach is able to control the adaptation. This was evaluated in a module constructed for the Neverwinter Nights game, the spiritual successor to the Baldur's Gate series [12], in the same manner as in the game of MiniGate. Results showed that the adaptation could be scaled so that the team using dynamic scripting won, for example 30% or 50% of the time.

**Benefits for the implementation** Though Spronck's approach is not directly applicable to Code Red: Triage because of its focus on enemy AI, the concept of determining which actions to be chosen based on earlier experience offers possibilities. Because a variety of adaptations can be made in games, the choice of adaptation is very important – selecting different adaptations and determining which are effective is a good possibility for doing so.

### 3.5.3 Agent Organizations

An agent is an autonomous entity which observes and acts upon an environment and directs its activity towards achieving goals (definition from [51]). A collaboration between Utrecht University and the Delft University of Technology carried out by Joost Westra on an agent-based approach to user adaptation in serious games, [61], has very fruitful possibilities for agent-based games. Their framework relies on the use of agents that have the roles of the non-player characters in the game. Their approach is meant to be decentralized through use of agents in these roles, as to cope with more complex scenarios in terms of possibly adaptable elements. This is the case because in the usual centralized approaches, defining the difficulty of all the sub-tasks is only feasible if there is a small number of possible adaptations. Instead, an agent organization of NPCs could be given a goal, e.g., aiding the player character with performing his task, after which they would use the actions they have available to complete their goal. Westra et al.'s framework is meant to monitor the agent actions, to direct them towards helping the player and to adapt where necessary. Thus, when using this framework, the designer of a serious game only needs to specify certain conditions on the adaptation without specifying the exact implementations. These are carried out in a different layer, viz. one in which the game world and beliefs of the agents are instantiated.

[8] A computer role-playing game (RPG) based on the Dungeons & Dragons tabletop RPG.

**Benefits for the implementation** Westra et al. acknowledge that their approach only becomes more useful than centralized approaches in more complex environments, especially when the history of actions of NPCs should be taken into account. This is not the case for Code Red: Triage, because it consists of a relatively simple scenario with static NPCs.

### 3.5.4 Player Modelling

When properties of a game are to be adapted to specific qualities of a player, one could say that the adaptation is inherent to a model of that player. Likewise, adaptation can already occur before a game is started, based on the player model. For instance, in an intelligent tutoring system called *AnimalWatch*, there was no difference in progress between genders, but there was a difference in how each of the genders preferred to solve math problems, see [4, 7].

So far, some early implementations have been made, as Yun et al. have shown with their framework called the Profile-based Adaptive Difficulty System (PADS), [63]. PADS takes a player model as its input, after which it updates this model during the game. The player model is an indication of the skill level of the player combined with the type of player, which signified if he was a challenge seeker or a victory seeker or a mix of the two.[9] Then, together with performance data from the game, the model leads to the possibility of updating the difficulty of the game.

In an experiment conducted by Yun et al. that implemented PADS in a third person shooting game, the experimenters found positive results. Their participants were players with varying (little, moderate and much) experience as well as having different play types, viz. victory seekers and challenge seeker. The subjects played three different versions of the game, one easy, one difficult and one automatic, adapting, version. With the help of an in-game survey, they discovered that players playing the adapted mode believed to be playing longer at the moderate difficulty level than they actually were – on average, they believed they were playing at moderate difficulty (with the alternatives being easy and hard) 73% of the time and at hard difficulty about 7% of the time. Respectively, the real fractions of time spent in these difficulties were 53% and 41%, which seems to reflect the effect of the participants being in the *flow*, because they *thought* they were playing at a difficulty level that was neither too hard nor too easy for the largest part of the time. Additionally, subjects were asked which mode they preferred, to which 87% responded that the automatic mode was their favourite. Furthermore, the in-game survey data showed that the participants experienced enjoyment 13% longer in the automatic version than in the static versions.

Another approach to player modelling for the use of adaptation that is a generalization of Yun et al.'s player type was devised by Charles et al. in [13]. They expressed their concerns with preconstructing a framework for the player model and suggested the use of a factorial model to form a more complex and higher order profile. Such a profile would be a breakdown of certain important characteristics, e.g., in an adventure game, stealthiness and combat ability, and assign numerical values to these characters. For instance, if a player favours stealth, but does not shy away from battle, he would have a profile of (0.7, 0.5). Regrettably, Charles et al. leave open the methods to implement this idea any further.

Related to player modelling, some work has been invested in the monitoring of player's emotions. Also led by Yun, research was performed on adapting a game through use of a so-called StressCam, [62]. This setup includes a camera that monitors the player's face and is able to detect various levels of stress, after which the game can adapt the difficulty level. For instance, if a player continues to struggle through a part of the game, his stress level will rise and this will become apparent in his facial expression. The game can then lower the difficulty. Similar efforts were made in the form of a gamepad that detects how much pressure is applied to its buttons to estimate how the player is faring, [55]. In their discussion about using frustration in video games, Gilleade & Dix, [24], detail some more proposals. All of these takes on dealing with player's emotions could well have their assets, yet because of time constraints this possibility is left for future research.

[9] Cf. sections 3.4 and 3.3 on player types and 'zones'.

**Benefits for the implementation** Using a player model can be a very insightful approach to adaptation, because it gives abstracted data about the player that can also be further analysed. It seems that, for adaptation, using a player model is inherent in adapting based on statistics garnered from gameplay. This is the case because a collection of statistics of a player can be seen as a player model. As in Jameson's general user-adaptive system schema, see section 3.3, a model is abstracted from the data, after which it is used to make decisions regarding the system. There are two reasons not to implement an explicit player model, the first of which amounts to the inclusion of an implicit player model in the form of gameplay data. The second is that because adaptation in Code Red: Triage is to be relatively simple, there is no need of an elaborate player model.

Implementing some way of dealing with players' emotions could prove to be fruitful, yet it is left for future research.

### 3.5.5 Active Adaptivity

For his master thesis at the University of Southern California, Jenova Chen designed a game that incorporated Csíkszentmihályi's flow in a game called *flOw*, see [14, 15, 16, 17]. The essence of Chen's project was to enable players of flOw to choose their own flow by actively adjusting the difficulty of the game, as opposed to having a passive software system that would be capable of doing the same.

flOw is a simple game in which the player controls a creature that has to be steered through various levels, defeating enemies that attack him and receiving upgrades for his creature so that combat may be easier. Each successive level is a bit more difficult than the previous one, yet the player can choose when to enter to next level or, if he finds it necessary to upgrade before progressing, to enter the previous level. In this manner, the player adapts the game to his needs while it ensues, which is reminiscent of the Montessori Method, as explained in section 3.3. However, this implementation is not so much adaptive as it is *adaptable*, i.e., the adjustment is user-controlled and not automated, which is in conflict with the subject of this thesis – creating an adaptive learning system which has automated responses to user behaviour.

**Benefits for the implementation** It should be noted that Chen's approach is not adaptive, but adaptable in nature: the player chooses his own challenge. Though this can be beneficial to other research, it opposes the goal of this thesis, namely implementing an *automated* difficulty adjustment.

# 4 Design of Adaptation

This section discusses the design for the implementation of dynamic complexity adaptation in Code Red: Triage.

## 4.1 The Basic Idea

In the previous section, various approaches to adaptation in games and learning systems were discussed, which contribute to the implementation in the following ways. The basic idea for adaptation in Code Red: Triage is the scaling of the difficulty level in order to create an optimal, i.e., not too low, nor too high, challenge, coinciding with the concept of flow from Csíkszentmihályi Mihály. This should lead to gradual increase of the player's skill as well as an increase in the challenge for that player, as Figure 11 (repeated from section 3.2) illustrates.

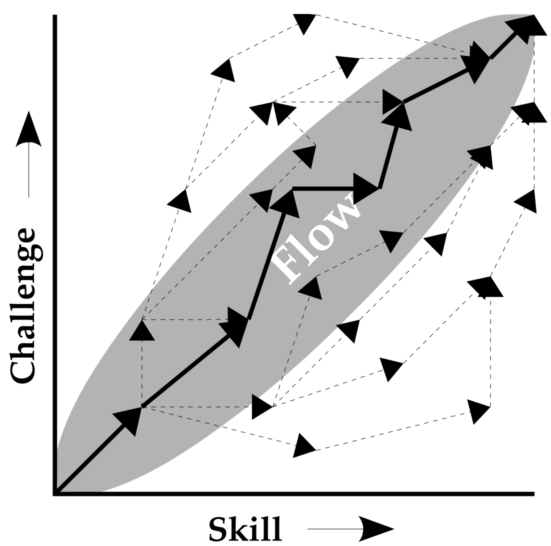



**Figure 11:** *Adaptation guides a player (bold arrows) in order to keep them 'in the flow', adapted from [15, p. 32].*

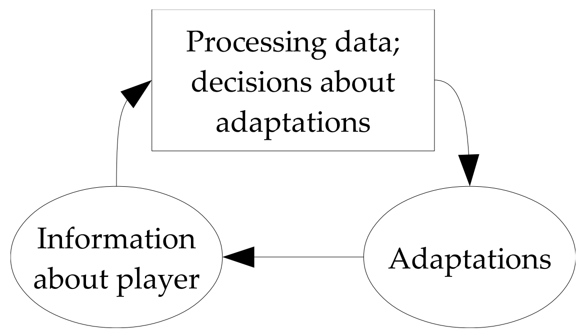



**Figure 12:** *Schema for adaptation in Code Red: Triage. Ellipses stand for input or output, the rectangle for processing methods.*

To perform such adaptations, Jameson's general user-adaptive system schema is taken as a basis for the framework, yet because there is no explicit need for a player model (see section 3.5.4), it is simplified. This model can be seen in Figure 12.

Abstractly, it can be asserted that the combination of the flow theory and the above feedback loop leads to a form of dynamic difficulty adaptation similar to those discussed in the previous section. The goal of this design is to increase the effectiveness of learning in a serious game. The instantiations of the various processes and information from Figure 12 for Code Red: Triage are detailed in the next section, where different alternatives are considered and, ultimately, chosen.

## 4.2 Implementation Alternatives for Adaptation

Bailey's idea, [6], of dividing a game into different attributes[10] to adapt seemed promising, but the inconcrete borders of these attributes make this approach more problematic than it should be, as dis-

[10] Player character, non-player character, game world/level and puzzle/obstacle attributes.

cussed in section 3.5.1. Instead, the different elements that can be adapted can be addressed without classification just as well. For starters, the experimental nature of this thesis restricts the possible alternatives, because it is best to keep as much conditions the same as possible. This makes for a better inter-condition comparison, so when each alternative is discussed, the implications for the experiment should be noted.

### 4.2.1 Location of Victims

Whereas the main task of players consists of performing triages on the victims, their initial task is finding the victims in the first place. Adapting the location of the victims may help the player find the victims and assist him with triaging all the victims within the time limit. However, results from earlier experiments pointed out that practically all players of Code Red: Triage were able to find all victims in the level. Therefore, implementing this adaptation would only be of use if the time limit would be shortened or the number of victims increased. At this point, there is no incentive to perform either of these alterations, so location adaptation is not implemented.

### 4.2.2 Complexity of Victims

Each of the victims a player triages is in a certain state – the triage procedure is used to infer this state. Logically, the complexity of the procedure depends on the state of the victim. The definition of complexity used in this thesis is as follows.

**Complexity of a victim:** The minimal number of sequential steps it takes to successfully triage a victim.

On first sight, it is not completely far-fetched to use the terms 'complexity' and 'difficulty' as interchangeable. However, a hypothetical case may exist in which one may find a certain sequence of triage steps more easy to complete correctly than another sequence, while the complexity of the first sequence is higher, i.e., it has more steps, than that of the latter. Keeping the experiment in mind, it can be asserted that complexity is an objective property, as it is defined as a fixed number of steps. This is opposed to difficulty, which is highly subjective. Therefore, the term 'complexity of victims' is used as a variable instead of their difficulty for the adaptation in CRT.

In order to help the player train triaging, the complexity level of the victim that is inspected by the player can be adapted to the skill level of the player. That is, based on the performance of the player during previous triage(s), an adaptation is made so that the next victim the player triages provides the correct challenge. E.g., if a player performs below a certain threshold, the next victim has a lower complexity than the previous one and inverse.

Despite the use of complexity instead of difficulty, this approach can still be seen as a form of dynamic difficulty adaptation, as Bailey and Hunicke and Chapman, [6, 29, 30], and several game developers (see section 3.4) implemented as well.

### 4.2.3 Remaining Alternatives

Because of the focus on the experiment that is performed, the different elements that are left for adaptation have few possibilities. Changing the speed of the player would be a difficult variable to consider when analysing the results. This would influence the difficulty of finding the victims, which is better refrained from, because it is the complexity of the victims themselves which is important.

Changing the scoring algorithm might be an efficient adaptation in games to keep them entertaining for different players, but for an objective training game, this is out of the question. This leaves environmental cueing and showing hints as remaining options for adaptation.

Early results from research by Van der Spek et al., [44], indicate that environmental cueing does not have a positive effect on the effectiveness of learning of players. Following their results, implementing environmental cueing is refrained from. Hints were already provided in the previous version of the game (see section 2), giving the player information about his performance, e.g., whether he missed some procedural steps or how fast he was. These hints could be adapted to provide less or more detailed information, but the score the player achieves is inherent with the steps he makes. Therefore, he should always receive feedback, else his possibility of learning could be obstructed.

## 4.3 Providing Victims of Correct Complexity

The adaptation that is implemented in Code Red: Triage relies on the game – or, rather, the *adaptation module*, the part of the game that performs the adaptations – letting the player triage victims that provide an apt challenge.

### 4.3.1 Dynamic Victim Set

Several difficulties have to be overcome for this approach to be successful. The first is that there is a *static* set of 19 victims: their number and respective complexities do not change over time. It may occur that the player performs above the threshold for adaptation each time, resulting in him performing triage on the most difficult victims quite soon. Then, when he has treated all complex victims in the set, only less complex victims will be left for him to triage. This will disrupt his flow, because the challenge presented to him will dwindle.

This issue can be surmounted by a combination of the following two changes to the set of victims.

1. The set is no longer static in terms of complexity, i.e., the contents of the set are subject to change. Because adaptation module chooses a victim with a correct challenge for the player, the victims that are presented to the player can be repeated.

2. The set is no longer static in terms of size, i.e., the player may continue playing the game until he has performed a successful triage on a victim from each different complexity level.

However, this solution is also prone to some complications, because in the case of (1), the adaptation can always adapt in the correct manner to the player's performance, but players may not arrive at a point in which they triage the more complex victims. Approach (2) solves this problem by letting the player keep performing triages until he is able to successfully complete the most complex cases, but this complicates the implementation of the victims of the level, because there can be an indefinite amount of victims. However, results obtained by earlier experiments performed by Van der Spek et al., [45], showed that most participants were able to perform correct triages on the most difficult victims in case of the static victim set used. Thus, the set of victims does not have to be expanded.

In effect, this design has two alternative scenarios, the first being that a player successfully – i.e., above a certain threshold – performs a triage on one of the most complex victims before he has triaged all the victims. At this point, his ability to perform this triage indicates that he has had sufficient training. The second scenario is that he does not attain this skill level when playing the game and triages all 19 victims.

### 4.3.2 Determining the Next Victim

Two other problems are related to the presence of the victims in the game world: the first is that the adaptation module needs to determine which victim the player will triage next to adapt its complexity. The second is that, when changing the victim properties, i.e., its complexity, the appearance of the victim has to change as well, e.g., whether it is standing upright or lying down and whether it has visible injuries. These two difficulties can be overcome by implementing some changes to the scenario in the metro level.

One approach to this is the modification of the existing metro level or creation of a new level in such a way that the player is only able to see one victim at a time. When this is set up so that the player can only move in one direction, e.g., the new level consists of a series of hallways, the adaptation module can determine which victim will be triaged next. However, this could lead to a lower level of immersion of the player, because Code Red: Triage should be situated in a somewhat realistic metro station. Therefore, this idea is not pursued any further.

An alternative approach has to do with the appearance of the victims in the previous version of the game, which coincided with their complexity, i.e., with their injuries. For instance, if a victim was heavily wounded, the in-game model would have recognizable injuries – while being realistic on one hand, this gave players prior knowledge before performing the triage on the other hand. This could lead the player to skip certain steps in the triage procedure, because he would already think he knew the information that would be gathered by those steps. For instance, if a victim was standing when the player approached him, he could infer a priori that the victim was still mobile, which should be checked as the first step of the procedure.

In essence, a choice is to be made between realism on one side, i.e., corresponding postures and animations for each victim, and reducing the influence of prior knowledge as much as possible, e.g., inferring that the victim is still mobile when he is standing, on the other side. Yet because Code Red: Triage is about learning – that is, the attainment of knowledge – reducing the influence of prior knowledge has a higher priority than maintaining realism and immersion.[11] Therefore, it would be a better option to have all victims assume the same posture. Placing the victims in a lying position seems to be the best alternative, because it leaves the option open whether the victim is still mobile. By implementing this change in the metro level, the problem of changing appearances of victims can be overcome and, moreover, the player may no longer infer a priori that the victim is mobile or not. Concurrently, the adaptation module can simply determine the victim case that is to be presented to the player at the moment the player selects to do a triage. In this way, there are no problems for the adaptation regarding the next victim the player will triage.

However, letting all victims assume the same posture has one drawback, which becomes clear in the case of a dead victim. The normal lying models have an animation loop that causes them to breath and move a little, which should not be the case with dead victims. Therefore, when the adaptation module determines which victim should be triaged and selects a victim that is supposed to be dead, the model animation will change from lying down and breathing to lying still. This is the better alternative to simply letting the victim model continue breathing or even letting all victims remain motionless. The latter decreases the possibility of immersion of the player severely while the former makes the information the player receives from the game contradictory. These assessments and the conclusions drawn from them are necessary, but may influence the outcome of the experiment, so side-effects of this choice have to be kept in mind.

#### 4.3.3 Tiering

What is important for the adaptive version of the game is that the skill level of the player needs to be induced. A straightforward approach to this is to determine for each victim whether the player performs above a certain threshold, i.e., above a certain score, after which the game can adapt the next victim accordingly.[12] The first version of Code Red: Triage already featured a scoring element that enabled players to see how well they performed each triage (see section 2), which can be used for this cause.

By dividing the victims in *tiers* corresponding to their complexity, the adaptation can be made more transparent. In the next section, these tiers are described in their implemented form. For instance, if the player performs above the required threshold, the adaptation module checks which tier the victim belonged to, after which it provides a victim of the following, higher tier. That is, following the definition of complexity from section 4.2.2, a victim which takes more steps to successfully triage.

If the player performs below the specified threshold, the next victim he attends to will be of the same level as the previous one. The algorithm will pick the next victim from each tier until all victims

[11] To a respectable limit, of course, because Code Red: Triage is still meant to be a serious game.

[12] The next section treats the implementation of the design discussed here and provides explicit details about the threshold value.

in that tier have been triaged – thereafter, a victim is selected from the successive tier. The choice to go up one tier even if the player does not have the required skill level is based on two things. First, for the alternative to occur, there would be need of extra victim cases or the repeating of victim cases which the player already triaged, of which the latter could be confusing for the player. Second, it could be possible that the player never passes the threshold value, no matter how many victim cases he triages. Then, the tier would only be repeated over and over. When the tiers are kept at fixed sizes, experiments can determine whether adaptive learning is more efficient than static learning, which is the research question studied in this thesis. Furthermore, data from Van der Spek et al.'s research, [45], showed that a good majority of the players were able to triage at least one of the victims of each tier correctly.

Thus, for the game to be completed by the player, he is required to reach the most complex tier and perform a successful triage on one of the victims contained therein or to triage all victims in that tier in case he is unable to triage one of them correctly. See Figure 13 for an example of this adaptation mechanic.

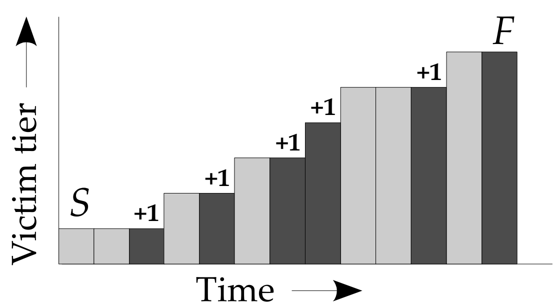



**Figure 13:** *Example sketch of the tier progression the adaptation algorithm could provide.*
S *and* F *denote the start and finish of the game. The dark grey bars indicate a successful triage;* +1 *stands for the increase of the tier for the next victim.*

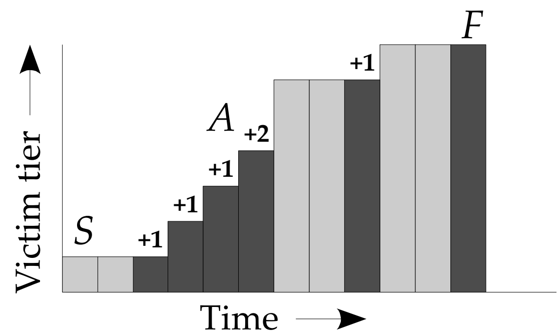



**Figure 14:** *Example sketch of the tier progression the adaptation algorithm could provide when* second-order adaptivity *is implemented.*
S *and* F *denote the start and finish of the game. The dark grey bars indicate a successful triage;* +1 *stands for the increase of the tier for the next victim.* A *denotes the point at which, after three successful successive triages, the next victim is two tiers higher than the previous victim (*+2*).*

Figure 13 provides an idea of how the adaptation algorithm should adapt the complexity of the victims over time during gameplay. In the beginning, the player performs an unsuccessful triage on a victim of the lowest tier, so the next victim the module selects is of the same tier (the second rectangle is as high as the first one). He successfully triages the third victim, so the next victim is of the second tier. Gameplay continues until the player performs a successful triage at *F*, after which no next victim is presented and the game ends.

Furthermore, the adaptation can also be adapted itself. The motivation for this second-order adaptivity is that if a player exhibits a steep learning curve, the adaptation can be adjusted to provide victims of *more than* one tier difference from the player's current tier. In Figure 14, an example of such an adjustment can be seen. This form of adaptation could make the learning process faster because the intermediate steps would be skipped. However, the danger of this adaptation is that such large steps can be too coarse again, so the module should at least have the possibility of returning to its normal behaviour of in- or decreasing the difficulty with one level. Because the adaptive version of Code Red: Triage only features a relatively small number of tiers for such an approach and, moreover, does not feature decreasing the difficulty, it is better to refrain from second-order adaptivity. Adaptive adaptivity would be more beneficial for frameworks in which smaller and more frequent adaptations can be made.

## 4.4 The Adaptation Loop

Summarizing, the design of the implementation of dynamic complexity adaptation is as follows. A set of 19 victims is placed lying in the metro level. When a player chooses to triage a victim, the adaptation module chooses a victim that is of the appropriate complexity for the player. To do so, the victims are divided into tiers corresponding to their complexity, which stems from the minimal number of steps the player has to make to triage the victim.

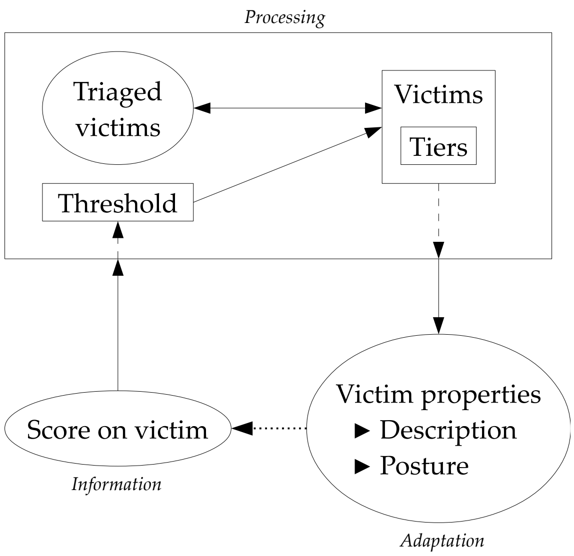



**Figure 15:** *Schema for adaptation in Code Red: Triage. Ellipses stand for input or output, the rectangle for processing methods. The dotted arrow indicates that the adaptation is repeated after each triage.*

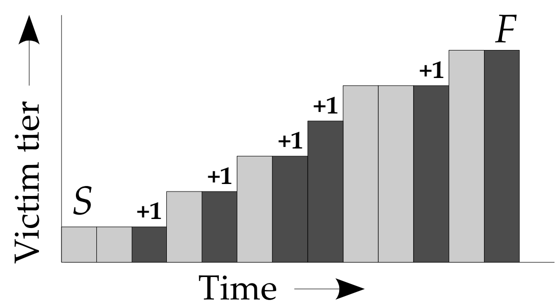



**Figure 16:** *Example sketch of the tier progression the adaptation algorithm could provide, repeated from Figure 13.*
S *and* F *denote the start and finish of the game. The dark grey squares indicate an increase of complexity from the previous victim.*

In Figure 15, the simplified form of Jameson user-adaptive system schema (see Figure 12) is instantiated for the adaptation in Code Red: Triage. Based on the *information* the adaptation module receives, namely the score of the player reaches and the tier the victim is in, it *processes* the information. This is done by comparing the score to a threshold value after which a victim is picked from a particular tier. If the player's score is higher than the threshold, the next victim will be of a higher tier, otherwise, it will be of the same tier. The adaptation module also checks which victims have already been triaged, so that the player does not re-triage them. When the last remaining victim of a tier has been triaged, the module selects the first victim from the following, higher tier to the player as the next victim. When the processing steps have been performed, a new victim is presented to the player. Additionally, when the victim description indicates that the victim is dead, its posture is modified so that the in-game model of the victim does not move anymore, but lies still in the level. These actions constitute the concrete *adaptation* of the game. The adaptation module loops through this schema until one of the end conditions has been fulfilled. These are (1) the player has performed a triage on a victim from the highest tier with a score higher than the threshold value and (2) when he has triaged all the victims from the highest tier.

As explained in more detail in the previous section, the algorithm lets the player progress through the game in a way akin to that of Figure 16.

# 5 Implementation of Adaptation

The previous section laid out the basic ideas for adaptivity in Code Red: Triage. This section details the specific instantiations of the victim set and the algorithm properties.

## 5.1 Victim Set

The victim set that was used by Van der Spek et al. in their experiments consists of 19 victim cases, which are re-used in this design. These cases can be categorized in six tiers by distinguishing the the sequences of steps or *paths* the player has to take in the triage procedure to determine the correct classification.[13] The different paths are listed in Table 1. They are arranged in order from shortest to longest path, i.e., in ascending complexity. Paths of equal length that include actions – as opposed to paths consisting only of checks – are considered to be more complex.

| **Tier** | **Path** |
|---|---|
| 1 | Mobility |
| 2 | Mobility, airway, breathing frequency |
| 3 | Mobility, airway, *chin lift* |
| 4 | Mobility, airway, breathing frequency, CRT |
| 5 | Mobility, airway, *jaw thrust*, *side position* |
| 6 | Mobility, airway, breathing frequency, CRT, pulse |

**Table 1:** *The six different paths the player has to take to classify the victims. Slanted words indicate actions.*

Additionally, the original in-game descriptions of the victims used by Van der Spek et al. were adjusted to exclude references to the in-game models, such as the sex of the victim.

## 5.2 Determining the Threshold Value

The adaptation that is designed for Code Red: Triage relies on the score the player reaches when he has triaged a victim. Depending on this score, the adaptation module decides if the player should enter the next tier of victims or stay in the same tier. Accordingly, the threshold value should indicate the minimum skill level the player should attain before advancing to a more complex tier.

To infer a correct threshold value, data from Van der Spek et al.'s research can be used. They performed a number of experiments in Code Red: Triage in which two properties of the game were varied, viz. (1) the number of buttons that was visible in the triage screen and (2) the complexity of the victim cases. Two instantiations were evaluated per property – for the number of buttons, these were (1a) a variable number of buttons, based on the difficulty of the victim (few buttons for the simple cases, more for the more complex cases), and (1b) the complete button set for each victim. The complexity of the victim cases was offered in two varieties, namely in (2a) a progressive fashion, where players could follow a 'logical' route through the level that ascertained a gradual increase

[13] These steps are explained in section 2.

in complexity, and (2b) varying as much as possible. These factors were combined to create four different game types and 53 participants in total took part in the experiments.

The choice to use one of these conditions as a reference for the threshold value in the adaptive version of the game relies on the condition that is used as the control condition in the experiment. This is the case because the experiment's goal is to determine the difference in efficiency between adaptive and non-adaptive learning. Thus, it is best to use an average score from previous experiments for that condition for the adaptive version, so that players may climb up a tier when they reach a score that players of the non-adaptive version reached on average. It follows that at this point, an explicit choice has to be made for the comparative condition.

In its most basic form, Code Red: Triage features a level in which victims are located so that the most straightforward path the player takes leads him to the victims in order of increasing difficulty, i.e., the progressive complexity (2a) condition. The flow of the adaptive version of the game will be very alike to this condition, with the exception that the player may skip certain victims when the adaptation module selects a next victim. The button property is the same in the adaptive version as in the non-adaptive version of the game for the experiment, so an arbitrary choice can be made here, for it does not influence the comparison of the two versions. Because it can easily be implemented, the complete button set (1b) condition is chosen.

For each distinct tier, an average score on all victims of that tier can be calculated per participant. These averages are then averaged again, but this time over all participants, so that an overall average of a score on a victim for a particular tier is determined. Using this method, the average scores of the 14 participants of the progressive complexity and complete button set condition can be computed, which are shown in Table 2.

| Tier | Average score | SD |
|---|---|---|
| 1 | 44.05 | 7.31 |
| 2 | 13.81 | 9.87 |
| 3 | 30.07 | 12.53 |
| 4 | 28.38 | 12.40 |
| 5 | 29.93 | 20.51 |
| 6 | 23.98 | 10.42 |

**Table 2:** *Average scores per tier for the progressive complexity & complete button set condition.*

| Tier | Threshold value |
|---|---|
| 1 | 44 |
| 2 | 14 |
| 3 | 30 |
| 4 | 28 |
| 5 | 30 |
| 6 | 24 |

**Table 3:** *Threshold values for the tiers.*

What is apparent is that the average score of the first tier is the highest, which is logical, because the complexity of the victims in this tier is the lowest. In this case, during the triage only one step needs to be taken together with a correct classification to successfully triage these victims. The average score of the second tier is a lot lower than that of the first, which can be attributed to the rise in steps that need to be taken during the triage, viz. from one to three in respectively the first and second tier. That there is a drop in the average score indicates that the participants experienced the victims from tier 2 as more difficult and. In fact, the average score on victims from tier 2 is the lowest of all tiers. However, this is no incentive to re-order the tiers based on these scores, i.e., from high to low scores – first tier 1, then tier 3, then tier 5, etc. This is the case because the order in which the victims are triaged influences how much the participants learn. For instance, if the victims from tier 2 were presented to the participant after all the other victims, the average score on the former would be much higher than is the case in Table 2. Therefore, no change is made to the design of the adaptation thus far and, rounding to the nearest integer values, the average scores from Table 2 become the threshold values shown in Table 3.

## 5.3 The Adaptation Module

Implementing adaptation into the existing source code took some effort, particularly because of static references, e.g., the victim names in the level. These had to be intercepted for the adaptive version to make the game choose different victims. Programming code was written for determining the tier of the previous victim and determining the next victim based on the score and the tier of the previous victim. At each new victim, it was checked whether his state was dead, so the posture of the victim in the game could correspond to that state.

In Listing 1 below, pseudocode is shown which describes how the module would act when a victim was selected to be triaged. Listing 2 shows pseudocode of the check that was made directly after a victim was triaged, to determine whether the game should end.

The only change made in the game levels was the posture of the victims, which would all be lying down. As Listing 1 indicates, when a selected victim was supposed to be dead, his posture would change accordingly.

```
// Check of which tier the previous victim was
int previoustier = tier of previous victim

// The first victim is always "s1"
string nextvictim = "s1"

// Check if the previous triage was successful
if ( score on previous victim >= threshold of previoustier )
  nextvictim = first untriaged victim from previoustier + 1
else
  if ( all victims from previoustier have been triaged )
    nextvictim = first untriaged victim from previoustier + 1
  else
    nextvictim = first untriaged victim from previoustier

// Check if the next victim should be dead and set his posture to dead
if ( nextvictim is a victim who is dead )
  set dead posture in game
```

**Listing 1:** *When a new victim is selected to be triaged, the adaptation module selects one of a suitable complexity.*

```
// Check of which tier the previous victim was
int previoustier = tier of previous victim

// After the victim has been triaged, check whether the game should end
if ( previoustier == 6
     && previoustier score on previous victim >= threshold
     || all victims from tier 6 have been triaged )
  end the game
```

**Listing 2:** *When a victim has been triaged, a check is made whether the game should end after that victim.*

# 6 Experiment

The research question this thesis seeks to answer is how adaptivity affects learning. In the previous sections (4 and 5), an implementation of dynamic complexity adaptation in Code Red: Triage was detailed. This section treats the experiment that was carried out to investigate the difference between adaptive and static learning.

## 6.1 Experimental Conditions

The experiment is comparative in nature, therefore the participants were randomly divided into two groups: one control group and one 'adaptive' group. The latter played the version of the game in which the DCA algorithm is implemented (the adaptive condition) and the former played the version in which the victims are static and divided over the map so that the most straightforward path establishes a progressive complexity (the control condition). Both versions have been construed to be as alike as possible where the complexity adaptation is not affected. That is, the victim postures are the same, to ensure that players do not have prior knowledge about the victim status (see section 4). Both conditions feature the same amount of buttons in the triage panel as well as the same descriptions for the victims.

## 6.2 Hypotheses

The introduction of this thesis formulated a short hypothesis to the research question whether adaptive learning is more efficient than static learning. With the performed work in mind, a more elaborate hypothesis can be fleshed out. First, two definitions of efficiency can be produced, which are detailed below.

**Time efficiency:** The amount of knowledge gained per time unit.

**Case efficiency:** The amount of knowledge gained per victim case.

The need for these two forms of efficiency stems from the possibility that time and victim cases may not have a linear relationship. For instance, a participant of the adaptive condition may spend the same time as one from the control condition to complete the game while triaging less victims. Then, when both participants have learned equally much from the game, it can be said that the adaptive version of the game is more case efficient, but both versions are equally time efficient.

These definitions lead to the revised versions of the hypotheses, which are investigated in this experiment.

**Hypothesis 1:** Adaptive learning is more efficient than static learning – that is, players of the adaptive version of the game gain more knowledge per victim or per second than players of the non-adaptive version.

**Hypothesis 2:** Adaptivity keeps the player in the flow at each moment in time, so the game should be more engaging to the players of the adaptive version of Code Red: Triage.

## 6.3 Participants

The participant group consisted mostly of students – with only three exceptions – totalling 28 participants. Not surprisingly, the mean age was 22.86 with a standard deviation of 5.68. 19 participants were male and 9 female. There were only 2 participants that indicated they never played computer games; 16 participants had some familiarity with computer games and 10 participants called themselves a gamer.

The participants were randomly and evenly divided in each of the two conditions so that 14 persons participated in each of them.

## 6.4 Measures

To measure the efficiency of the players' learning and their engagement, four types of instruments were used, listed below.

The *in-game scores* formed the first measure: an indication of the progression of the player in the game. Statistics from the game that were logged included triaged victims, number of triaged victims, tier of victim, time per victim, total time, score per victim and total score.

Second, a computer program called *Pathfinder* was used, [53], to measure the mental model of players. It lets users rate pairs of words based on their relation to each other. In this manner, it is able to elicit one's mental model and draw a graph in which words are connected to each other and spaced based on their assigned relatedness. In this experiment, a small set of concepts associated with the triage procedure is presented to the participant. As Van der Spek et al. noted, [44], a larger portion of concepts creates a combinatorial explosion of pairs the participant has to rate, making it tiresome for him and unreliable. This mental model can then be compared to an average of the mental models of domain experts. By performing this test before and after the game is played, the change in knowledge can be inferred and used as another measure. This is an important measure, because the mental model is used to attain a sufficient level of situational awareness in risk management tasks, see [34]. For example, 'mobility' and 'T3' would be closely related and 'mobility' and 'chin lift' would not be very related. Figure 17 shows the referent model Van der Spek et al. created, [43], by abstracting it from several domain experts.

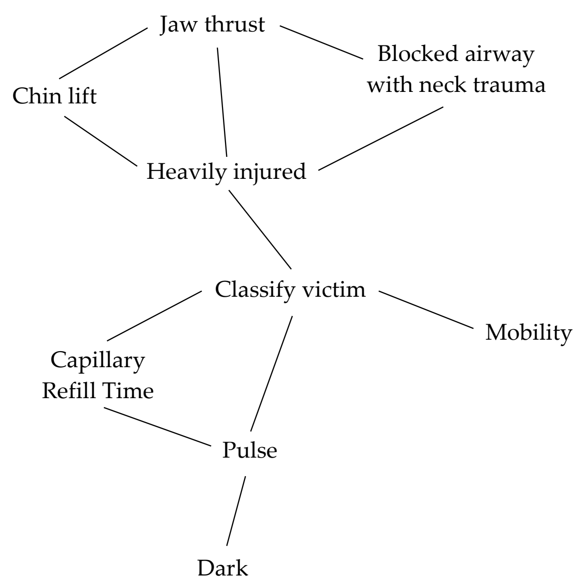



**Figure 17:** *Referent mental model, elicited from domain experts.*

Both a pre- and a post-*knowledge assessment questionnaire* are given to each participant to determine their prior and posterior knowledge about the triage procedure. By comparing the differences in answers between the two questionnaires, the change in knowledge could be determined. These questionnaires contained 8 text and 8 visual items that covered instances of victim cases from each of the tiers as described in section 5 as well as conceptual questions. The distinction between these two was made to analyse whether the visual nature of the game would influence the procedural knowledge – in that case, participants should perform better on the visual items. Figures 18 & 19 give translated examples of the items; see Appendix A for the complete questionnaires the participants received.

The fourth measure was a subset of the ITC-SOPI questionnaire, see [37], which was used as an *engagement questionnaire*. The ITC-SOPI questionnaire is a Likert-type scale questionnaire aimed at

In the primary triage, a victim with a respiratory frequency that is too high will probably be classified as:

A. T1

B. T2

C. T3

D. Dead

**Figure 18:** *Example text item from questionnaire, translated.*

Indicate which action should be performed first according to the primary triage.

27-year old man. It is very dark. The victim is not able to walk and has no problems breathing.


**Figure 19:** *Example visual item from questionnaire, translated. Participants were asked to indicate which action they should perform next according to the primary triage.*

the amount of immersion (or 'presence') a person experiences in multimedia. The mean score of the twelve items that were used for this experiment indicates the level of immersion the participant experienced. Two questions were included about the difficulty of the game and the amount of fun the participants experienced. Three questions concerning adaptivity and relating to *flow* were also integrated, of which translated versions are listed below.

1. "I think I have had sufficient training."
2. "At no moment in time, the game was too easy."
3. "At no moment in time, the game was too hard."

These questions featured in the engagement questionnaires of both participant groups and each questionnaire consisted of a total of 17 items. See Appendix B for the complete engagement questionnaire.

## 6.5 Apparatus

As described in section 2, the game was made with the Source SDK, a free kit designed by Valve Software to produce mods for Source-engine games. Adaptivity was implemented in the game as described in section 5. It was played on a 17" laptop at a resolution of $1920 \times 1200$ pixels with circumaural headphones in a room with the lights turned off. The laptop and the participant were shielded from further outside influences by a large blue screen, see Figure 20, as to improve immersion. The graphic settings were set at their maximum and the game ran at a constant 60 frames per second.

## 6.6 Procedure

Having taken place behind the blue screen, the participants were asked to fill out a short demographics questionnaire. Hereafter, they took part in determining their mental model through use of the Pathfinder software. Then, the participants answered a questionnaire consisting of both textual and visual items to determine their prior knowledge about the triage procedure.

Before playing the game, the participants were given instructions about Code Red: Triage and were informed about its goal. Nothing was revealed to them about the condition they took part in. When they felt they were ready for the task, the lights in the room were shut off and the participants were asked to put on the headphones as to make optimal immersion in the game possible. They could then commence the game. Playing the game from start to finish took each participant at most 25 minutes: a few minutes for the entry level, a few more for the hallway part and a maximum of 17 minutes for the metro platform part, in which the triages took place.

The scores participants reached in the game gave information about their performance. In the adaptive condition, data was gathered about the number of victims each participant triaged. Directly after the participants finished playing the game, they were asked to fill out the engagement questionnaire. Then, as before, their mental model was elicited thanks to the Pathfinder software and they were asked to answer the knowledge questionnaire again with the questions in a different order.

Finally, the participants were thanked for their cooperation and they received a coupon for their work.

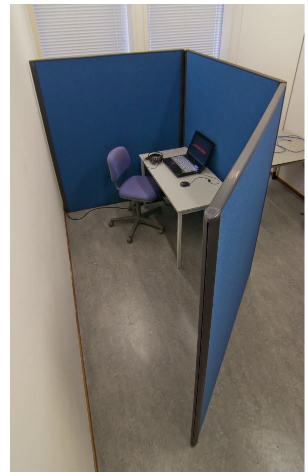


**Figure 20:** *The setup for the experiment: the laptop, the headset and the blue screen.*

## 6.7 Results

This subsection is devoted to the results from the four different measures from section 6.4.

### 6.7.1 Game Results

Logged statistics from the game were analysed to determine whether there was a difference in time and number of victims between the adaptive condition and the control condition. Table 4 shows the means of the total time, the total number of triaged victim and the total score per condition. The control group played the version of the game in which they were required to find all 19 victims. Three persons in this group did not achieve this goal, therefore the mean of the number of triaged victims of this group is lower than 19.

| | **Control group** | | **Adaptive group** | |
|---|---|---|---|---|
| | *M* | *SD* | *M* | *SD* |
| *Total time (s)* | 746.29 | 128.90 | 525.21 | 108.23 |
| *Triaged victims (out of 19)* | 18.64 | 0.75 | 10.71 | 2.09 |
| *In-game score (out of 1900)* | 777.71 | 321.21 | 316.43 | 107.85 |

**Table 4:** *Means of total, total number of triaged victims and total in-game score.*

A one-way ANOVA was carried out on these three variables separately, with the condition as a fixed factor and each variable as dependent variable. The total time it took participants to complete the entire game differed significantly between the two conditions, $F(1,26) = 24.15$, $p < 0.001$, where it took the participants of the adaptive group less time than those of the control group. The same holds for the total number of victims participants had triaged at the end of the game, $F(1,26) = 178.57$, $p <$

0.001, with the adaptive group performing less triages than the control group. These two results indicate that the implementation of adaptivity in Code Red: Triage had definite effects for the gameplay, viz. shortening the game. This is the first step in the direction of confirming or disprove hypothesis 1 from section 6.2. In combination with the data from the knowledge assessment questionnaires and the Pathfinder program, the learning efficiency can be determined. Furthermore, there is a high level of correlation between the number of triaged victims and the total time spent, $r(26) = 0.69$, $p < 0.001$. This is an indication that the efficiency definitions from section 6.2 would probably not yield different results in terms of the efficiency of adaptive learning versus that of static learning.

First, however, the in-game score has to be analysed. An ANOVA (in-game score as dependent variable; condition as fixed factor) shows that the total score at the end of the game differed significantly between the conditions as well, $F(1,26) = 25.95$, $p < 0.001$, but this can be attributed to the fact that there was a difference in the number of triaged victims. Therefore, the average score per second and average score per victim were calculated, which are shown in Table 5.

| | **Control group** | | **Adaptive group** | |
|---|---|---|---|---|
| | *M* | *SD* | *M* | *SD* |
| *Avg. in-game score per second* | 1.06 | 0.43 | 0.62 | 0.21 |
| *Avg. in-game score per victim* | 41.79 | 17.16 | 31.04 | 12.47 |

**Table 5:** *Means of average in-game score per second and per victim.*

The average in-game scores per second and victim were taken as dependent variables in a one-way ANOVA with the condition as a fixed factor as well. Obtained results still indicate that there was significant difference between the conditions regarding the average score per second, $F(1,26) = 11.83$, $p = 0.002$, with the control group outperforming the adaptive group. A difference that approached significant values was found for the average score per victim, with the control group performing better than the adaptive group, namely $F(1,26) = 3.60$, $p = 0.07$. Nonetheless, it appears that in the game the adaptive group performed, on average, worse than the control group. An explanation for this is straightforward: for a participant from the adaptive group to advance to the next tier, he is required to reach a score above a certain threshold, as explained in section 5. This leads to the possibility that this person would, for example, only triage two victims from tier 1, after which he goes on triaging victims from tier 2. Opposed to the adaptive group, participants from the control group were required to triage all victims from each tier. In the case of tier 1, they would triage a total of four victims. Assuming that a participant from the control group reaches a score above the threshold that holds for participants from the adaptive group on the second victim as well, he still has to triage two more victims from that tier. He can then apply the knowledge gained from the first two victims on the two victims that remain, from which it follows that he may reach higher scores on these. Thus, the average score per victim for that tier would be higher for the control group than for the adaptive group.

Because adaptivity in Code Red: Triage was meant to let participants perform on a particular level, it can be hypothesized that their scores would not differentiate much per victim. Figure 21 shows the means of the victim scores both conditions reached per victim. That is, victim 1 is the first victim that was triaged by a participant, victim 2 the second, etc. This implies that, for instance, victim number 8 is not the same for all participants. Yet because these are means of those scores, the average progress per condition can be studied.

What is obvious is that the control group scores much higher on the first few victims, as the first four are all of the same complexity. The adaptive group, however, continues to victims of tier 2, i.e., with a higher complexity, after two or three victims, which explains the lower average scores on their

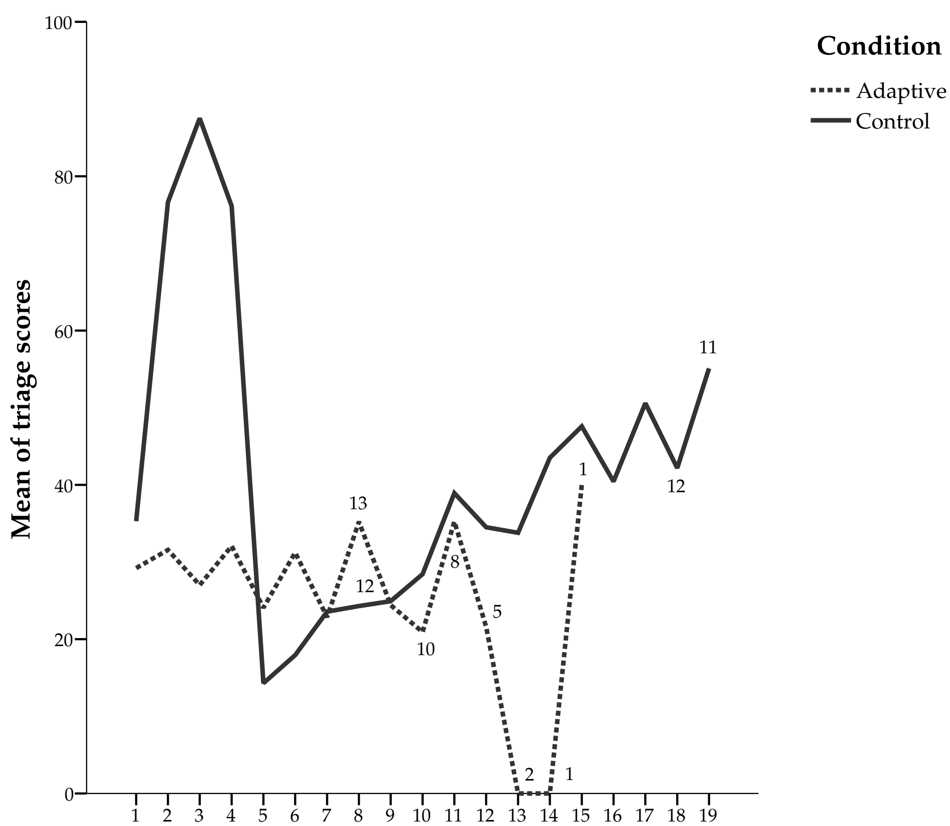



**Figure 21:** *Means of triage scores, in the order in which the victims were triaged. All values were calculated by taking the means of all 14 participants of that group, unless otherwise indicated.*

part. What is also apparent is that the adaptive group stays on more or less the same 'level': their average scores do not fluctuate much, but stay around 30. Participants from the control group seem to perform better over time, as the inclination of average scores from 5 to 19 shows. Yet because the adaptive group finished the game after approximately 11 victims, there is little to compare between groups. Nonetheless, the result that the adaptive condition yields a more constant score is in line with the concept of flow, because the game is at least never too easy, otherwise the participants would have score much higher.

### 6.7.2 Mental Model Results

The Pathfinder software returns a similarity score between 0 (completely different) and 1 (perfectly the same), which indicates the likeness of the participant's mental model to that of the domain experts, i.e., the *mental model score*.[14] This can be considered as the 'raw' score a participant reaches on the test. To analyse the efficiency of adaptive learning, the similarity score that was deduced from the participants after the game was played was divided through the number of victims each respective participant triaged and time it took each participant to complete the game. These variables can then be interpreted as a measure for efficiency, viz. knowledge per victim or per second. The means of these variables are shown in Table 6.

[14] This value is corrected for chance, which entails that the corrected similarity scores can reach from a value less than 0 to a value less than 1.

| | Control group | | Adaptive group | |
|---|---|---|---|---|
| | *M* | *SD* | *M* | *SD* |
| *Pre-test mental model score* | 0.043 | 0.098 | 0.101 | 0.071 |
| *Post-test mental model score* | 0.281 | 0.071 | 0.271 | 0.185 |
| *Post-test mental model score divided by time* | 0.000384 | 0.000103 | 0.000513 | 0.000390 |
| *Post-test mental model score divided by no. victims* | 0.0151 | 0.0041 | 0.0277 | 0.0190 |

**Table 6:** *Means of mental model scores.*

Instead of taking the increase of the mental model score, i.e., by subtracting the pre-test value from the post-test value, an ANCOVA was used to determine the difference between the conditions. The condition was taken as a fixed factor, the post-test values as the dependent variable and the pre-test values as a covariate, so that the ANCOVA adjusts for the pre-test. This approach yielded that there was no significant difference between the conditions ($F(1,25) < 1$), which indicates that both groups have improved their mental model by the same amount.

Most importantly, combining this result with the results from the previous section – where a significant difference in time and triaged victims was found – leads to the confirmation of the hypothesis that the adaptive group may have learned more efficiently. The similarity scores divided by the time ("Per second" in Table 6) it took to complete the game and the number of victims that were triaged ("Per victim" in Table 6) can be analysed through ANCOVAs as well. Taking the "Per second" score as the dependent variable, condition as a fixed factor and the pre-test variable as a covariate again, no significant difference was found between conditions, $F(1,25) = 1.04$, $p = 0.32$. However, there was one participant in the adaptive group whose "Per second" value was removed more than two standard deviations from the mean. This outlier was removed from the data set and another ANCOVA (with the same parameters) was performed, yet this did not return a significant difference as well, $F(1,25) = 3.39$, $p = 0.08$. From this, it must be concluded that, when efficiency is defined as gained knowledge per second, adaptive learning can not be said to be significantly more efficient. However, the difference between conditions approaches conventional significance, which could imply that participants who played the game longer and performed not so good on the mental model test were responsible for the deviating values.

The other definition of efficiency – gained knowledge per victim – returns a result that is definitely significant. An ANCOVA (dependent variable: "Per victim" variable; fixed factor: condition; covariate: pre-test value) reveals a significant difference between the two conditions, $F(1,25) = 5.05$, $p = 0.03$. This implies that adaptive learning can be assumed to be significantly more case efficient than static learning.

#### 6.7.3 Knowledge Assessment Results

Similar to the method use above, the answers to the knowledge assessment questionnaires (*KA* scores) were analysed. The raw scores are percentages of questions answered correctly, where the values per second and per victim are the raw scores divided by resp. the time it took the participant to complete the game and the number of victims he triaged. This was done separately for the text and visual items in the questionnaires to investigate whether the visual representation of the game influenced KA scores. The means from both conditions for these values are shown in Table 7.

An ANCOVA (dependent variable: post-test; fixed factor: condition; covariate: pre-test) was used

| | Text items | | | | Visual items | | | |
|---|---|---|---|---|---|---|---|---|
| | Control group | | Adaptive group | | Control group | | Adaptive group | |
| | *M* | *SD* | *M* | *SD* | *M* | *SD* | *M* | *SD* |
| *Pre-test KA scores* | 37.50 | 23.00 | 40.18 | 16.39 | 32.14 | 16.05 | 33.04 | 7.92 |
| *Post-test KA scores* | 66.01 | 23.73 | 58.93 | 19.26 | 66.07 | 22.70 | 73.21 | 15.39 |
| *Post-test KA scores divided by time* | 0.89 | 0.032 | 0.115 | 0.038 | 0.088 | 0.025 | 0.143 | 0.031 |
| *Post-test KA scores divided by no. victims* | 3.56 | 1.30 | 5.67 | 2.02 | 3.56 | 1.26 | 7.11 | 2.07 |

**Table 7:** *Means of the knowledge assessment scores, presented as percentages of correctly answered items.*

to investigate the difference between the conditions concerning the absolute knowledge about the text items and visual items. Both tests returned an insignificant difference ($F(1,25) < 1$) on the raw post-test KA scores, implying that the adaptive group and the control group learned equally well from the game. ANOVAs with the post-test KA scores as the dependent variable and the type of items as fixed factor show that the differences between the scores on the text and visual items were not significant ($p > 0.05$). Therefore, the visual nature of the game did not influence the attaining of procedural knowledge.

To determine the time efficiency, an ANCOVA with the per second variable as the dependent variable was used for the text and visual items – fixed factor and covariate were the same as above. For both these items, significant differences were found; for the text items $F(1,25) = 3.56$, $p = 0.02$ and for the visual items $F(1,25) = 28.07$, $p < 0.001$. These results signify that, concerning procedural questions, adaptive learning is more efficient than static learning, as the means of the per second variables of the adaptive group are higher than those of the control group.

The same computation was performed on the per victim variable, viz. an ANCOVA with the per victim variable as the dependent variable – fixed factor and covariate still the same as above. Case efficiency turns out to be significantly higher for adaptive learning than for static learning, with $F(1,25) = 10.32$, $p = 0.004$ for the text items and $F(1,25) = 30.19$, $p < 0.001$ for the visual items.

### 6.7.4 Engagement Results

The engagement questionnaire gave some insight into how the participants experienced the game. Overall, they enjoyed the game, as the answers to the question of how much fun the game was indicated (for the control condition: $M = 7.43$, $SD = 0.94$; for the adaptive condition: $M = 7.14$, $SD = 0.66$). Nonetheless, they also experienced it to be quite difficult (control condition: $M = 7.07$, $SD = 1.49$; adaptive condition: $M = 7.57$, $SD = 1.70$). One-way ANOVAs (dependent variable: resp. answers to the fun and difficulty questions; fixed factor: condition) show no significant difference ($F(26) < 1$) between the conditions.

In Table 8 below, the means of ITC-SOPI scores and the questions to the three different questions concerning adaptivity are listed per condition.

Participants from both groups felt equally immersed in the game (one-way ANOVA with ITC-SOPI as dependent variable and condition as fixed factor, $F(26) < 1$). One question that was part of the ITC-SOPI questionnaire, viz. “I would have liked the gamed to lasted longer.”, produced

| | Control group | | Adaptive group | |
|---|---|---|---|---|
| | *M* | *SD* | *M* | *SD* |
| *ITC-SOPI* | 3.57 | 0.30 | 3.56 | 0.42 |
| *"I would have liked the gamed to lasted longer."* | 3.57 | 1.02 | 4.21 | 0.43 |
| *"I think I have had sufficient training."* | 2.29 | 1.20 | 2.07 | 1.21 |
| *"At no moment, the game was too easy."* | 3.21 | 1.37 | 4.07 | 1.21 |
| *"At no moment, the game was too hard."* | 2.57 | 0.94 | 2.93 | 1.14 |

**Table 8:** *Means of ITC-SOPI questionnaire and adaptivity-related questions. Questions could be answered on a 5-point Likert-type scale.*

significantly different means for the two conditions, with the adaptive group agreeing more on this statement than the control group. A one-way ANOVA with the answer as the dependent variable and the condition as a fixed factor returned $F(26) = 4.77$, $p = 0.04$. This is no surprise as the adaptive group finished the game in significantly shorter times than the control group.

One-way ANOVAs on the three questions with the respective answers as dependent variable and condition as fixed factor returned no significant result, resp. $F(26) < 1$; $F(26) = 3.09$, $p = 0.09$ and $F(26) < 1$. While the means of the two latter questions were higher for the adaptive group, there is no basis on which to assume that adaptivity in Code Red: Triage affects the experience of the game's difficulty.

Pertaining to engagement, it can be stated that it is not influenced by adaptivity, which disproves the second hypothesis of this experiment. Results found by Van der Spek in other experiments, [43], indicate mostly the same engagement scores, with no significant difference ($p > 0.05$) between groups. This could be an indication that serious games are not easily made more engaging or that the means to determine the engagement are simply not sufficient. Instruments such as the StressCam, see section 3.5.4 and [62], could possibly provide more fruitful results.

## 6.8 Conclusion

This experiment was conducted to investigate the hypothesis of whether adaptive learning is more efficient than static learning. Analysing the game statistics (total time, number of triaged victims, etc.), a significant difference in both total time and number of triaged victims was found between conditions. Participants of the adaptive condition completed the game in less time and with a lower number of performed triages than those of the control condition. Furthermore, the total time and number of victims proved to be strongly correlated, leading to the suggestion that the distinction between the two definitions of efficiency need not have been necessary.

Scrutiny of the mental model results (the Pathfinder data) and the knowledge assessment results (the questionnaires) revealed that there was no significant difference ($p > 0.05$) between both groups concerning total mental model construction and total procedural knowledge. As hypothesized, results yielded that adaptive learning was more efficient than static learning regarding procedural knowledge. Additionally, it was found that there was no significant difference in the increase of procedural knowledge between the text and visual items in the questionnaires. Thus, the visual nature of the game did not influence the participants' learning.

A similar result was found when analysing the mental model results, where adaptive learning showed to be more *case efficient* than static learning. The difference in *time efficiency* only approached significant values. This discrepancy could be related to the larger standard deviation in the adaptive group, as the scores of the adaptive group were much more varied than those of the control group, thus influencing the outcome of the analysis.

Lastly, the participants found the game to be quite engaging, with no significant difference ($p > 0.05$) in the ITC-SOPI test score between groups. The only thing thing that was commented on by the adaptive group that was different from the answers of the control group was the time they spent playing the game, which they felt was too short.

Overall, it can safely be assumed that the hypothesis concerning efficiency is confirmed. In-game adaptivity makes learning more efficient than static learning. It can not be concluded that adaptivity makes the game more engaging, as was hypothesized.

# 7 General Conclusion & Discussion

The research question of this thesis was whether an educational game could be made adaptive to its players, so that they might learn more efficiently. The performed research indicates that the first hypothesis – adaptive learning is more efficient than static learning – is indeed validated. The second hypothesis, which asserted that players would be more engaged by the adaptive version, was not confirmed.

In section 2, the game, called *Code Red: Triage*, was described in full: a training simulation for the triage procedure, constructed by Erik van der Spek, [43]. Players of this game assume the role of a medical first responder whose task it is to find and *triage* victims of a terrorist strike. Triaging victims happens through selecting a number of actions and checks the player can perform on those victims, after which he is required to assign a priority to them. The game gives feedback on the performed triages and the player can use this feedback to update his thoughts about the procedure, which he should then apply to the next victim.

Different approaches to adaptivity in education and games were discussed in section 3, where the concept of *flow* was introduced. Flow is the notion of being in a state that offers sufficient challenge for a particular skill level. This seemed to relate to adaptivity in such a way that adaptivity can cause a person to be 'in the flow' by tailoring the difficulty of a task to the person's abilities. Furthermore, several researchers performed studies into the adaptation of (educational) games, where some adaptations were dialogue systems that provided hints and others adaptations to fighting or shooting games. One concept, viz. *dynamic difficulty adjustment*, could be used in a general way, for it is synonymous with the adaptation of difficulty. In the game, this would translate to the adaptation of the *complexity* of the next victim – the minimum number of steps necessary to successfully triage a victim – that the player triages, as a victim's difficulty is subjective and one's complexity is not.

Sections 4 and 5 were devoted to fleshing out and implementing this idea. A feedback loop was used to govern the construction of adaptation in Code Red: Triage, which consisted of (1) obtaining information from the game, (2) processing this information and (3) adapting the game. It was decided that the adaptation would be used in an experiment which would determine whether adaptive learning would be more efficient than static learning. Therefore, several choices were made in the design stadium, among which the decision to build the adaptive version of the game in such a way that, in the 'worst' case – in which players would learn very slowly – its gameplay would be identical to that of the static version. This would enable the experiment to be more straightforward in determining which version would empower the player to learn more efficiently.

Thus, the experiment was set up so that two groups participated: one playing the adaptive version and one playing the non-adaptive version (the control group), as section 6 described. Results obtained from the 28 participants yielded that, on average, both groups learned equally well when time and number of triaged victims were not taken into account. Their mental models had improved by the same amount, as did their scores on the procedural questionnaires. Yet a significant difference ($p < 0.001$) was found between groups in the amount of time and the total number of victims the participants triaged, in the advantage of the adaptive version of Code Red: Triage. On average, participants of the adaptive group spent approximately 30% shorter in the game than those of the control group. They also performed, on average, 11 triages, where the control group performed 19, yet their knowledge after the game differed non-significantly from the knowledge of the control group. Thus, the adaptive group was able to gain the same amount of knowledge in a shorter time span with less triaged victims, which indicates that the implementation of adaptivity allowed players to gain knowledge more efficiently. This also holds for the improvement of their mental model, yet for the time efficiency the difference only approached conventional significance levels due to outliers in the adaptive group.

The experiment confirmed the hypothesis concerning efficiency: adaptive learning is more efficient than static learning in Code Red: Triage. By adapting the victims the player triages to his performance, time spent in the game as well as the amount of victims that were triaged were decreased significantly.

The hypothesis that the adaptive version of the game would be more engaging was disproved through comparison of the ITC-SOPI questionnaire results. No significant difference ($p > 0.05$) was found between conditions, which could indicate that engagement is hard to affect or that measuring the engagement through a questionnaire simply not suffices. It seems that influencing the players in the discussed way to get them 'in the flow' does not have the expected effects. This could be the case because flow by itself is a rather abstract concept that may not be measured in terms of an in-game score or through questionnaires. Other instruments that are more direct, e.g., stress meters, or in-game questionnaires, could possibly yield more precise results for determining the player's state. In term, these results could be used to influence the adaptation mechanism in order to construe a better way of letting the player enter the flow.

Concluding, this thesis has shown an instantiation of adaptation in an educational game called Code Red: Triage, based on the progress of the player through the game. It was shown that by doing so, learning was made more efficient, which is a good incentive for including adaptation in more educational games. As a result, training procedures may be made more efficient, saving time and effort on the participants' side as well as on the designers' side. Concerning engagement, more research should be carried out to discover methods to adapt games in order to make players feel more engaged.

## Future Research

Because several choices were made during the design and implementation process, this thesis leaves other alternatives open for continued research. For instance, the set of buttons that is presented to players during triages could be adapted or, more simply, increased over time, starting with only one button for the first victim. This was already done by Van der Spek, [43], yet not in conjunction with adaptivity as was implemented in the version of Code Red: Triage that was used for this thesis. Perchance this additional adaptation would make players learn even more efficiently.

Work could also be performed on adapting the feedback that is given to players. This could happen in the form of more detailed critique on the player's triage, for instance by stating more explicitly what went right and what went wrong. During the triage, if the player seems to be having trouble deciding which actions to take, hints could be provided. However, these possibilities should be monitored carefully, because the game still needs to be interactive – otherwise it would scarcely be more than an on-screen demonstration that pin-points which actions one should take at every step of a triage.

The measures used for the experiment performed for this thesis only include post-tests that are given directly after the game is played. Further research could be spent on testing the long-term memory of participants to determine how much this is affected by adaptive learning.

As this thesis only investigates an implementation of adaptivity in an educational game that is meant to learn players a particular procedure, future research could delve into different forms of educational games, e.g., war simulations and naval simulations. The first steps have already been taken (see section 3), but more empirical research is necessary to investigate which parts of adaptivity can contribute to these simulations and which will not.

Lastly, another option for making training procedures adapt themselves to their participants is the notion of *adaptability*, as briefly discussed in section 3.5.5. In this form of adaptation, the *participants* indicate which direction they wish to travel in, e.g., whether they want to practice one part of the procedure or repeat another part. This approach also has its cons, because participants may not know what is best for them. Yet, as said, this is left for future research to determine.

# References


[1] Elektra Project. http://www.elektra-project.org, February 2011.

[2] Ernest Adams. Difficulty Modes and Dynamic Difficulty Adjustment. http://www.designersnotebook.com/Columns/095_Difficulty_Modes/095_difficulty_modes.htm, May 2008.

[3] Vincent Aleven and Kenneth R. Koedinger. An effective metacognitive strategy: learning by doing and explaining with a computer-based Cognitive Tutor. *Cognitive Science*, 26(2):147–179, April 2002.

[4] I. Arroyo, J. E. Beck, B. P. Woolf, C. R. Beal, and K. Schultz. Microadapting AnimalWatch to gender and cognitive differences with respect to hint interactivity and symbolism. In *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, 2000.

[5] Beatrice A. Ashem and Ernest G. Poser. *Adaptive Learning: Behavior Modification with Children*. Pergamon Press, Oxford, 1973.

[6] Christine Bailey and Michael Katchabaw. An experimental testbed to enable autodynamic difficulty in modern video games. In *Proceedings of the 2005 GameOn North America Conference*, 2005.

[7] C. Beal, J. Beck, D. Westbrook, M. Atkin, and P. Cohen. Intelligent modeling of the user in interactive entertainment. In *AAAI Stanford Spring Symposium, Stanford CA*, 2002.

[8] Bethesda Game Studios. The Elder Scrolls III: Morrowind. PC, Xbox, 2002.

[9] Bethesda Game Studios. The Elder Scrolls IV: Oblivion. PC, Xbox 360, PlayStation 3, 2006.

[10] Bethesda Game Studios. FallOut 3. PC, Xbox 360, PlayStation 3, 2008.

[11] BioWare. Baldur's Gate. PC, 1998.

[12] BioWare. Neverwinter Nights. PC, 2002.

[13] D. Charles, A. Kerr, M. McNeill, M. McAlister, M. Black, J. Kcklich, A. Moore, and K. Stringer. Player-centred game design: Player modelling and adaptive digital games. In *Proceedings of the Digital Games Research Conference*, volume 285, 2005.

[14] Jenova Chen. Flow in Games. Master thesis, University of Southern California, 2006.

[15] Jenova Chen. Flow in games (and everything else). *Communications of the ACM*, 50(4):31–34, April 2007.

[16] Jenova Chen. flOw. http://www.jenovachen.com/flowingames/flowing.htm, February 2011.

[17] Jenova Chen. Flow in Games, A Jenova Chen MFA Thesis. http://www.jenovachen.com/flowingames/thesis.htm, February 2011.

[18] Clover Studio. God Hand. PlayStation 2, 2006.

[19] Mihály Csíkszentmihályi. *Flow: The Psychology of Optimal Experience*. Harper Perennial, March 1991.

[20] Mihály Csíkszentmihályi, S. Abuhamdeh, and J. Nakamura. Flow. In A. Elliot, editor, *Handbook of Competence and Motivation*, pages 598–698. New York: The Guilford Press, 2005.

[21] Steven Delrue. Case Closed. PC, 2001.

[22] Sidney D'Mello, Scotty Craig, Amy Witherspoon, Bethany McDaniel, and Arthur Graesser. Automatic detection of learner's affect from conversational cues. *User Modeling and User-Adapted Interaction*, 18(1):45–80, February 2008.

[23] EA Canada. The Need for Speed. 3DO, PC, PlayStation, Sega Saturn, 1994.

[24] K.M. Gilleade and A. Dix. Using Frustration in the Design of Adaptive Videogames. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology*, pages 228–232. ACM, 2004.

[25] Robert Glaser. *Adaptive education: Individual diversity and learning*. Holt, Rinehart and Winston, New York, 1977.

[26] A. C. Graesser, P. Chipman, B. C. Haynes, and A. Olney. AutoTutor: an intelligent tutoring system with mixed-initiative dialogue. *IEEE Transactions on Education*, 48(4):612–618, 2005.

[27] A.C. Graesser, G.T. Jackson, E.C. Mathews, H.H. Mitchell, A. Olney, M. Ventura, P. Chipman, D. Franceschetti, X. Hu, M.M. Louwerse, and N.K. Person. Why/AutoTutor: A test of learning gains from a physics tutor with natural language dialogue. *Proceedings of the 25th Annual Conference for Cognitive Science Society*, pages 1–6, 2003.

[28] T.J. Hodgetts. Triage: A Position Statement. *European Union Core Group on Disaster Medicine*, 2002.

[29] Robin Hunicke. The Case for Dynamic Difficulty Adjustment in Aames. In *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*, pages 429–433, 2005.

[30] Robin Hunicke and Vernell Chapman. AI for dynamic difficulty adjustment in games. In *Challenges in Game Artificial Intelligence AAAI Workshop*, pages 91–96, 2004.

[31] Anthony Jameson. Adaptive Interfaces and Agents. In Julie A. Jacko and Andrews Sears, editors, *The Human-Computer Interaction Handbook*, pages 305–330. Lawrence Erlbaum Associates, 2002.

[32] Steve Jarvis and Sara de Freitas. Evaluation of an Immersive Learning Programme to Support Triage Training. In *Proceedings of the 2009 Conference in Games and Virtual Worlds for Serious Applications*, VS-GAMES '09, pages 117–122. IEEE Computer Society, 2009.

[33] Michael Katchabaw, Derek Elliott, and Stephen Danton. Neomancer: An exercise in interdisciplinary academic game development. In *The Proceedings of the DiGRA 2005 Conference: Changing Views–Worlds in Play*, 2005.

[34] W. Kintsch. *Comprehension: A Paradigm for Cognition*. Cambridge University Press, 1998.

[35] T. K Landauer, P. W. Foltz, and D. Laham. An Introduction to Latent Semantic Analysis. *Discourse Processes*, 25:259–284, 1998.

[36] Legend Entertainment. Unreal II: The Awakening. PC, Xbox, 2001.

[37] J. Lessiter, J. Freeman, E. Keogh, and J. Davidoff. A Cross-media Presence Questionnaire: The ITC-Sense of Presence Inventory. *Presence: Teleoperators & Virtual Environments*, 10(3):282–297, 2001.

[38] David McClure. Adaptive Difficulty. `http://www.gamecareerguide.com/features/805/adaptive_.php`, April 2009.

[39] Scott Miller. Game Matters: Autodynamic difficulty. `http://dukenukem.typepad.com/game_matters/2004/01/autoadjusting_g.html`, January 2004.

[40] Neversoft. Guitar Hero 5. PlayStation 2, PlayStation 3, Xbox 360, Wii, 2009.

[41] Nintendo. Mario Kart. Super Nintendo Entertainment System, 1992.

[42] E.D. van der Spek. COgnition-based DEsign Rules Enhancing Decisionmaking TRaining In A Game Environment (Code Red Triage): doctoral consortium paper. In *Proceedings of the 28th Annual European Conference on Cognitive Ergonomics*, ECCE '10, pages 319–322. ACM, 2010.

[43] E.D. van der Spek. *Experiments in Serious Game Design*. PhD thesis, Utrecht University, 2011.

[44] E.D. van der Spek, H. van Oostendorp, P. Wouters, and L. Aarnoudse. Attentional Cueing in Serious Games. In *2010 Second International Conference on Games and Virtual Worlds for Serious Applications*, pages 119–125. IEEE, 2010.

[45] E.D. van der Spek, P.J.M. Wouters, and H. van Oostendorp. Code Red: Triage, Or, COgnition-based DEsign Rules Enhancing Decisionmaking TRaining In A Game Environment. *British Journal of Educational Technology*, 2010.

[46] Neil Peirce, Owen Conlan, and Vincent Wade. Adaptive Educational Games: Providing Non-invasive Personalised Learning Experiences. *Second IEEE International Conference on Digital Games and Intelligent Toys Based Education*, pages 28–35, November 2008.

[47] Russ Pitts. Oblivion: The Dagobah Cave. http://www.escapistmagazine.com/articles/view/editorials/op-ed/992-Oblivion-The-Dagobah-Cave, August 2006.

[48] K. Rathunde and Mihály Csíkszentmihályi. Middle school students' motivation and quality of experience: A comparison of Montessori and traditional school environments. *American Journal of Education*, 111(3):341–371, 2005.

[49] Remedy Entertainment. Max Payne. PC, 2001.

[50] Ritual Entertainment. Sin: Episodes. PC, 2006.

[51] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[52] Adam Saltsman. Game Changers: Dynamic Difficulty. http://www.gamasutra.com/blogs/AdamSaltsman/20090507/1340/Game_Changers_Dynamic_Difficulty.php, September 2009.

[53] R.W. Schvaneveldt. *Pathfinder Associative Networks: Studies in Knowledge Organization*. Ablex Publishing, 1990.

[54] Pieter Spronck. *Adaptive Game AI*. PhD. dissertation, Universiteit Maastricht, 2005.

[55] J. Sykes and S. Brown. Affective Gaming: Measuring Emotion Through the Gamepad. In *CHI '03 Extended Abstracts on Human factors in Computing Systems*, pages 732–733. ACM, 2003.

[56] Valve Corporation. Half-Life. PC, PlayStation 2, 1998.

[57] Valve Corporation. Half-Life 2. PC, PlayStation 3, Xbox, Xbox 360, 2004.

[58] Valve Corporation. Left 4 Dead. PC, Xbox 360, 2008.

[59] Valve Corporation. Left 4 Dead 2. PC, Xbox 360, 2009.

[60] Kurt Vanlehn, Randolph M. Jones, and Michelene T. H. Chi. A Model of the Self-Explanation Effect. *Journal of the Learning Sciences*, 2(1):1–59, 1992.

[61] Joost Westra, Frank Dignum, and Virginia Dignum. Guiding User Adaptation in Serious Games. *Agents for Games and Simulations II*, pages 117–131, 2011.

[62] C. Yun, D. Shastri, I. Pavlidis, and Z. Deng. O' Game, Can You Feel My Frustration?: Improving User's Gaming Experience via StressCam. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 2195–2204. ACM, 2009.

[63] C. Yun, P. Trevino, W. Holtkamp, and Z. Deng. PADS: Enhancing gaming experience using profile-based adaptive difficulty system. In *Proceedings of the 5th ACM SIGGRAPH Symposium on Video Games*, pages 31–36, 2010.

## Appendix A – Knowledge Assessment Questionnaire

## Voorkennistest – Omcirkel de letter bij het juiste antwoord.

1. Bij de primaire triage van een slachtoffer wordt eerst onderzocht:
   a. Of de ademweg geblokkeerd is
   b. Of het slachtoffer nog kan lopen
   c. Of er uitwendige bloedingen zijn
   d. Hoe hoog de hartslag is

2. Een slachtoffer met een te hoge ademhalingsfrequentie valt tijdens de primaire triage waarschijnlijk in de triage klasse:
   a. T1
   b. T2
   c. T3
   d. Dood

3. Op een vrijdagnacht heeft een ernstig ongeluk plaatsgevonden op een donkere provinciale weg. Bij een slachtoffer wilt u de bloedcirculatie meten. Welke methode kunt u hiervoor tijdens de primaire triage in deze situatie het best gebruiken?
   a. De capillaire refill tijd
   b. Het meten van de bloeddruk
   c. De polsslagmeting
   d. Geen van de methoden is geschikt onder deze omstandigheden

4. Een slachtoffer heeft een vrije ademweg. De ademweg was uit zichzelf al vrij, als hulpverlener hoefde u hiervoor niet in te grijpen. Wat is de volgende stap die u volgens de primaire triage procedure moet uitvoeren?
   a. De capillaire refill tijd bepalen
   b. De ademhalingsfrequentie bepalen
   c. Het slachtoffer in een stabiele zijligging leggen
   d. De bloeddruk meten

5. Bij de triage van een volwassen slachtoffer met een geblokkeerde ademweg hebt u een sterk vermoeden van een nektrauma. Welke behandeling is tijdens de primaire triage dan het best?

a. Een chin lift (kinlift) toepassen
b. In de stabiele zijligging leggen
c. Een jaw thrust toepassen
d. Een halskraag aanbrengen

6. Elke triage klasse heeft een eigen kleur. De kleur voor triage klasse T2 is:

a. Blauw
b. Rood
c. Groen
d. Geel

7. Beschouw de volgende stellingen:

I. Een slachtoffer dat na het vrij maken van de ademweg nog steeds niet zelfstandig kan ademhalen, valt in triage klasse T1.
II. De CRT wordt gemeten via het nagelbed.

Welk van de stellingen is juist?

a. Zowel I als II is juist
b. Zowel I als II is onjuist
c. Alleen I is juist
d. Alleen II is juist

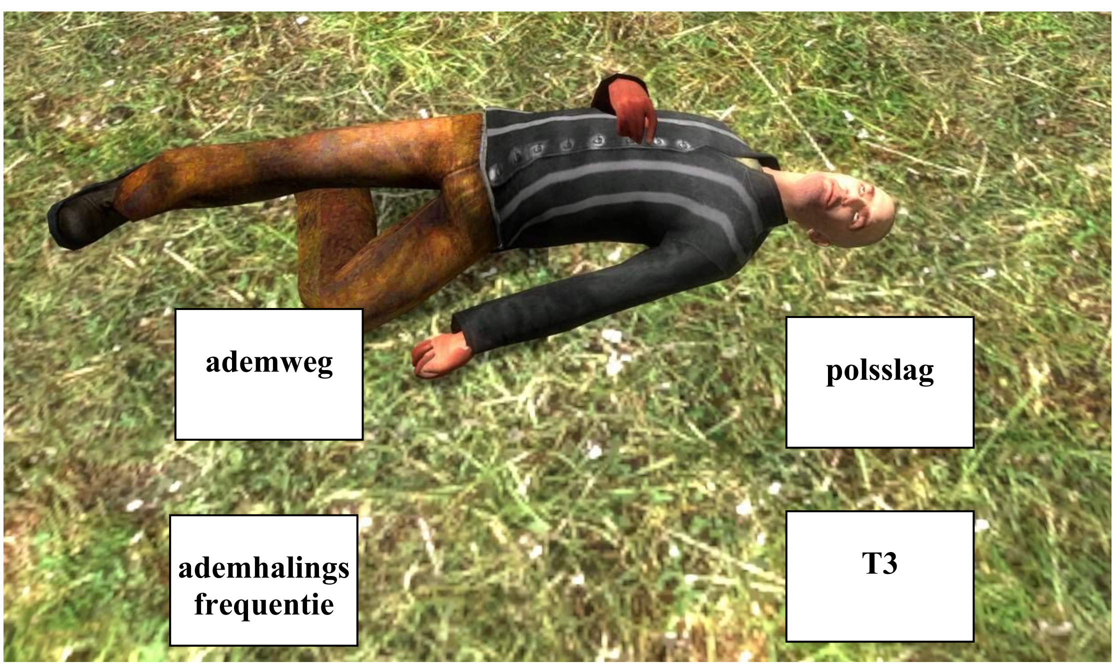
8. Bij een kettingbotsing treft u een slachtoffer met twee gebroken benen aan. Het slachtoffer kan zelfstandig ademhalen, heeft een normale ademhalingsfrequentie en een snelle capillaire refill tijd. Onder welke triage klasse zou u het slachtoffer indelen:

a. T1
b. T2
c. T3
d. Dood

**Visuele items – Kruis bij de volgende vragen steeds het vakje met de actie aan die volgens de procedure voor de primaire triage het eerste uitgevoerd dient te worden.**

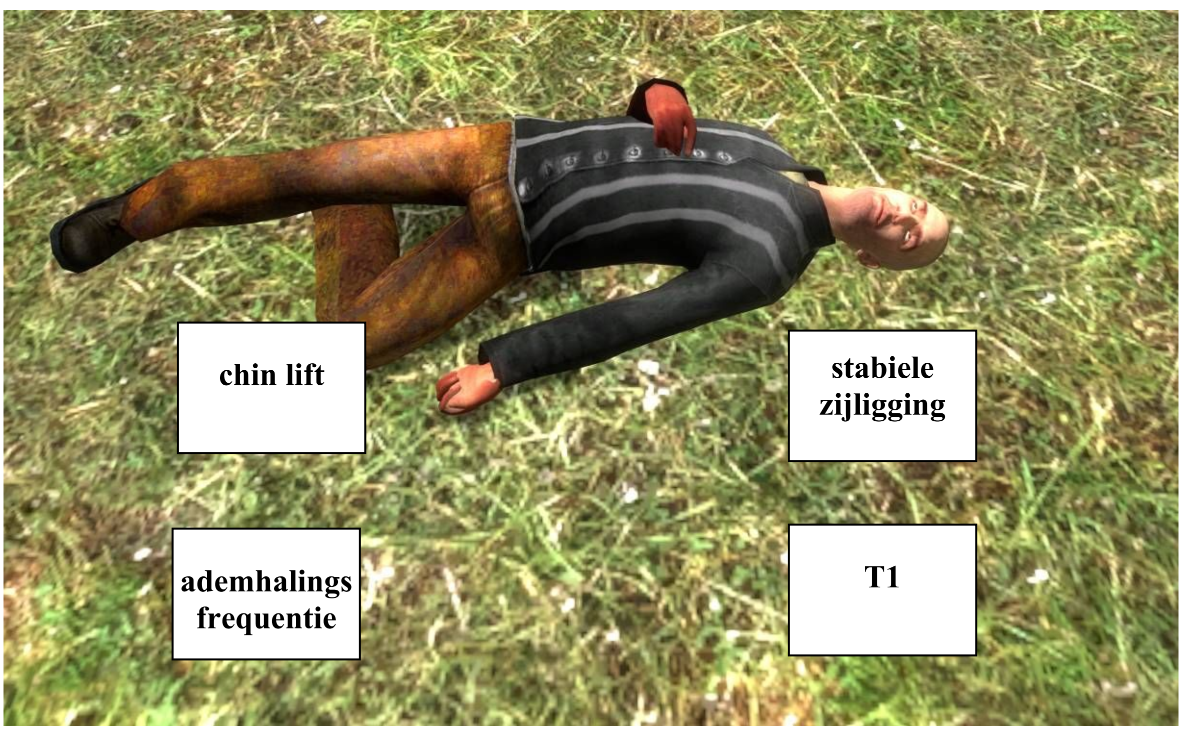
9. Man van 27 jaar. Het is erg donker. Het slachtoffer kan niet lopen en kan vrij ademhalen.


10. Man van 46. Er is genoeg daglicht voor een triage. De man is een beetje duizelig maar zegt dat hij nog kan lopen.



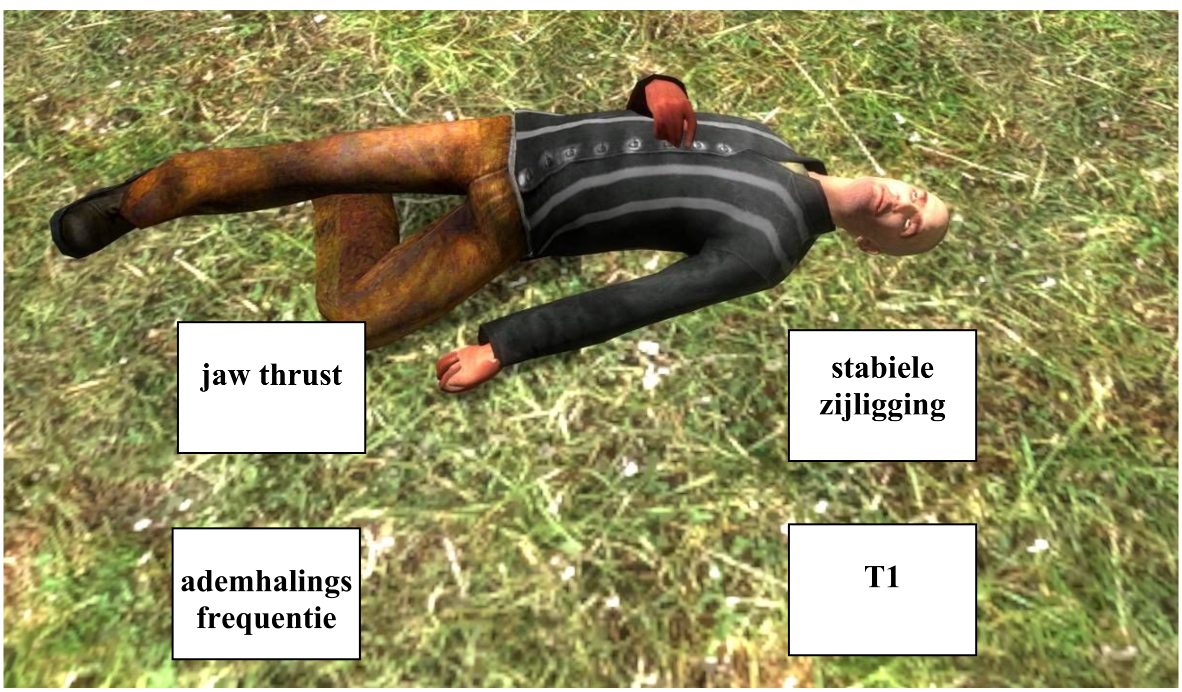
11. Man van 27 jaar. Genoeg licht voor een triage.


12. Man van 46. Er is genoeg daglicht voor een triage. De man heeft aangegeven dat hij niet kan lopen. Ook heeft u al een geblokkeerde ademweg geconstateerd.



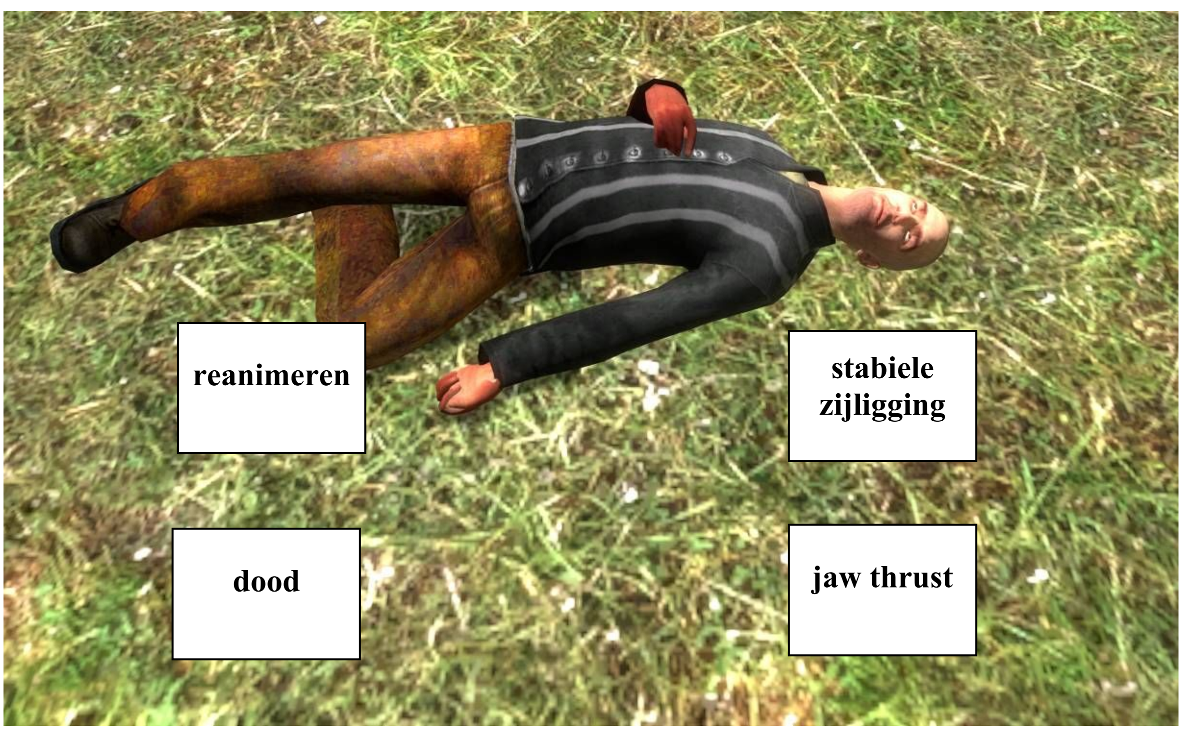
13. Man van 40 jaar. Het meten van de capillaire refill tijd is zojuist mislukt omdat het te donker is.


14. Man van 51. Er is genoeg daglicht voor een triage. De ademweg was geblokkeerd, maar u heeft deze zojuist vrijgemaakt en het slachtoffer ademt weer normaal.



15. Man van 23 jaar. Er is genoeg licht. Het slachtoffer is niet in staat te lopen en u heeft al bepaald dat zijn ademweg vrij is en zijn ademhalingsfrequentie binnen de normen valt.


16. Man van 30. Er is genoeg daglicht voor een triage. U heeft een geblokkeerde ademweg geconstateerd, maar nadat u deze vrij heeft gemaakt is de ademhaling niet op gang gekomen.



# Appendix B – Engagement Questionnaire

## Vragenlijst over hoe u de game ervaren heeft

(Wees alstublieft zo eerlijk mogelijk; we hebben niets aan sociaal wenselijke antwoorden!)

Op een schaal van 1 t/m 10 van makkelijk naar moeilijk, geef aan hoe moeilijk u het vond om een goede score te halen in de game.

| ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
|---|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **zeer makkelijk** | | | | | | | | | **zeer moeilijk** |

Op een schaal van 1 t/m 10, geef aan hoe leuk u de game vond.

| ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ | ☐ |
|---|---|---|---|---|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** |
| **zeer vervelend** | | | | | | | | | **zeer leuk** |

De volgende vragenlijst bestaat uit een aantal stellingen, welke worden gevolgd door een score van 1 tot en met 5. Hier moet u aan geven in welke mate u het eens bent met de stelling, waarbij 1 staat voor *helemaal mee oneens*, en 5 staat voor *helemaal mee eens*. Zet een kruisje in het vakje behorend bij de score die u zou willen geven.

| | *helemaal mee oneens* | *mee oneens* | *niet mee eens of oneens* | *mee eens* | *helemaal mee eens* |
|---|---|---|---|---|---|
| 1 Ik vind het jammer dat de game is afgelopen | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 2 Ik heb het gevoel teruggekomen te zijn van een avontuur | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 3 Ik zou het leuk gevonden hebben als de game wat langer duurde | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |

| | | | | | |
|---|---|---|---|---|---|
| 4 Ik kan me delen van de game levendig herinneren | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 5 De game was nooit te makkelijk | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 6 Ik voelde mezelf in de game wereld gezogen worden | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 7 Ik voelde me betrokken bij de game wereld | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 8 Ik denk dat ik voldoende training heb gehad | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 9 Ik was het gevoel van tijd verloren | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 10 Ik had plezier tijdens het spelen van de game | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 11 Mijn ervaring was intens | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 12 Ik had meer aandacht voor de game dan voor andere dingen die me bezig houden (zoals dagdromen) | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 13 De inhoud van de game sprak mij aan | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 14 De game was nooit te moeilijk | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |
| 15 Ik ervoer emoties tijdens het spelen | ☐ 1 | ☐ 2 | ☐ 3 | ☐ 4 | ☐ 5 |